Signal Processing and Machine Learning Techniques for Human Verification Based on Finger Textures

Raid Rafi Omar Al-Nima

Newcastle University

Newcastle Upon Tyne, UK

A thesis submitted for the degree of

Doctor of Philosophy

November 2017

NEWCASTLE UNIVERSITY

SCHOOL OF ENGINEERING

I, Raid Rafi Omar Al-Nima, confirm that this thesis with the presented work are my own achievements.

I have read and understand the penalties associated with plagiarism.

Signature:

Date:

SUPERVISOR'S CERTIFICATE

This is to certify that the entitled thesis "Signal Processing and Machine Learning Techniques for Human Verification Based on Finger Textures" has been prepared under my supervision at the School of Engineering / Newcastle University for the degree of PhD in Computer Engineering.

Signature:

Supervisor: Prof. Dr. Jonathon A. Chambers

Date:

Signature:

Student: Raid Rafi Omar Al-Nima

Date:

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

بَلَىٰ قَٰدِرِينَ عَلَىٰٓ أَن نُّسَوِّيَ بَنَانَهُۥ ﴿٤﴾

سُورَةُ القِيَامَةِ

Yes [we are] able [even] to proportion his[/her] finger.

Abstract

In recent years, Finger Textures (FTs) have attracted considerable attention as potential biometric characteristics. They can provide robust recognition performance as they have various human-specific features, such as wrinkles and apparent lines distributed along the inner surface of all fingers. The main topic of this thesis is verifying people according to their unique FT patterns by exploiting signal processing and machine learning techniques.

A Robust Finger Segmentation (RFS) method is first proposed to isolate finger images from a hand area. It is able to detect the fingers as objects from a hand image. An efficient adaptive finger segmentation method is also suggested to address the problem of alignment variations in the hand image called the Adaptive and Robust Finger Segmentation (ARFS) method.

A new Multi-scale Sobel Angles Local Binary Pattern (MSALBP) feature extraction method is proposed which combines the Sobel direction angles with the Multi-Scale Local Binary Pattern (MSLBP). Moreover, an enhanced method called the Enhanced Local Line Binary Pattern (ELLBP) is designed to efficiently analyse the FT patterns. As a result, a powerful human verification scheme based on finger Feature Level Fusion with a Probabilistic Neural Network (FLFPNN) is proposed. A multi-object fusion method, termed the Finger Contribution Fusion Neural Network (FCFNN), combines the contribution scores of the finger objects.

The verification performances are examined in the case of missing FT areas. Consequently, to overcome finger regions which are poorly imaged a method is suggested to salvage missing FT elements by exploiting the information embedded within the trained Probabilistic Neural Network (PNN). Finally, a novel method to produce a Receiver

Operating Characteristic (ROC) curve from a PNN is suggested. Furthermore, additional development to this method is applied to generate the ROC graph from the FCFNN.

Three databases are employed for evaluation: The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D), Indian Institute of Technology (IIT) Delhi and Spectral 460nm (S460) from the CASIA Multi-Spectral (CASIAMS) databases. Comparative simulation studies confirm the efficiency of the proposed methods for human verification.

The main advantage of both segmentation approaches, the RFS and ARFS, is that they can collect all the FT features. The best results have been benchmarked for the ELLBP feature extraction with the FCFNN, where the best Equal Error Rate (EER) values for the three databases PolyU3D2D, IIT Delhi and CASIAMS (S460) have been achieved 0.11%, 1.35% and 0%, respectively. The proposed salvage approach for the missing feature elements has the capability to enhance the verification performance for the FLFPNN. Moreover, ROC graphs have been successively established from the PNN and FCFNN.

Statement of Originality

The contributions of this thesis have been supported by different journal and conference papers, which have been generated during the journey of my study. They can be stated as follows:

[1] R. R. O. Al-Nima, M. A. M. Abdullah, M. T. S. Al-Kaltakchi, S. S. Dlay, W. L. Woo, and J. A. Chambers, "Finger Texture Biometric Verification Exploiting a Multi-scale Sobel Angles Local Binary Pattern and Score-based Fusion", *Elsevier, Digital Signal Processing*, vol. 70, 2017.

https://doi.org/10.1016/j.dsp.2017.08.002

The originality of this paper can be found in Chapter 5 and Chapter 6. That is, the Multi-scale Sobel Angles Local Binary Pattern (MSALBP) feature extraction approach is explained and evaluated in Chapter 5. Furthermore, the innovative fusion method named the Finger Contribution Fusion Neural Network (FCFNN) is described and examined in Chapter 6. Moreover, the Receiver Operating Characteristic (ROC) curves for the FCFNN are depicted in this paper and how these curves are produced is clarified in Appendix A.

[2] R. R. O. Al-Nima, S. S. Dlay, W. L. Woo and J. A. Chambers, "Efficient Finger Segmentation Robust to Hand Alignment in Imaging with Application to Human Verification," *5th IEEE International Workshop on Biometrics and Forensics (IWBF)*, 2017.

The contribution of this paper is a novel segmentation approach, which is illustrated in Chapter 4 under the name Adaptive and Robust Finger Segmentation (ARFS). In addition, an efficient finger fusion scheme is

explained in the same paper and included in Chapter 6 under the term Feature Level Fusion with Probabilistic Neural Network (FLFPNN).

[3] R. R. O. Al-Nima, S. S. Dlay, S. A. M. Al-Sumaidaee, W. L. Woo and J. A. Chambers, "Robust feature extraction and salvage schemes for finger texture based biometrics," in *IET Biometrics*, vol. 6, no. 2, 2017.
doi: 10.1049/iet-bmt.2016.0090.
In this paper four contributions are presented. Firstly, the Robust Finger Segmentation (RFS) approach, which can be found in Chapter 4. Secondly, the Enhanced Local Line Binary Pattern (ELLBP) feature extraction method, which can be seen in Chapter 5. Thirdly, evaluating the missing FT elements, which is included in Chapter 6. Fourthly, applying the salvage approach, which is involved in Chapter 6 too.

[4] R. R. O. Al-Nima, S. S. Dlay, W. L. Woo and J. A. Chambers, "A novel biometric approach to generate ROC curve from the Probabilistic Neural Network," *24th IEEE Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 2016.
doi: 10.1109/SIU.2016.7495697.
In this paper a new method to generate the ROC graph from a Probabilistic Neural Network (PNN) is introduced. The details of this method can be found in Appendix A.

[5] R. R. O. Al-Nima, S. S. Dlay, W. L. Woo and J. A. Chambers, "Human authentication with finger textures based on image feature enhancement," *2nd IET International Conference on Intelligent Signal Processing (ISP)*, London, 2015.
doi: 10.1049/cp.2015.1784.
This was my first contribution paper related to verifying people by using the Finger Texture (FT) characteristic. It has been utilized in Chapter 4 for comparisons.

Moreover, additional papers have been established during my study. They can be considered as extending supporting work and they indirectly relate to this thesis. These are as follows:

[6] S. A. M. Al-Sumaidaee, M. A. M. Abdullah, R. R. O. Al-Nima, S. S. Dlay and J. A. Chambers, "Multi-gradient Features and Elongated Quinary Pattern Encoding for Image-based Facial Expression Recognition," *Elsevier, Pattern Recognition*, vol. 71, 2017.

[7] M. A. M. Abdullah, R. R. Al-Nima, S. S. Dlay, W. L. Woo and J. A. Chambers, "Cross-spectral Iris Matching for Surveillance Applications," *Springer, Surveillance in Action Technologies for Civilian, Military and Cyber Surveillance*, Chapter 5, 2017.
https://doi.org/10.1007/978-3-319-68533-5_5.

[8] R. R. O. Al-Nima, M. T. S. Al-Kaltakchi, S. S. Dlay, W. L. Woo, and J. A. Chambers, "Personal Verification Based on Multi-Spectral Finger Texture Lighting Images", *In preparation for submission to IET Signal Processing*, 2017.

[9] R. R. O. Al-Nima, S. S. Dlay, W. L. Woo, "A New Approach to Predicting Physical Biometrics from Behavioural Biometrics," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, No. 11, 2014.

Acknowledgements

"My Lord, enable me to be grateful for Your favo[u]r which You have bestowed upon me and upon my parents and to work righteousness of which You will approve and make righteous for me my offspring. Indeed, I have repented to You, and indeed, I am of the Muslims." [1]
So, thanks a lot to you my God (((Allah))). Then, I want to thank all the people who positively supported me. Firstly, my parents; my main supervisor Prof. Jonathon Chambers; my second supervisor Prof. Satnam Dlay; my third supervisor Dr. Wai Woo; the ministry of higher education and scientific research in Iraq (MOHESR); the Technical college of Mosul; the Iraqi Cultural Attaché; the active people in the MOHESR, who strongly supported Iraqi students; Thukaa, Ahmed, Raiq, Mamdooh and Rawnaq Rafi; Lumia, Duaa and Mohammed Khalid; Diaa, Hanaa, Basheer, Nazar, Shifa, Talal and Maqbool Ali; Maysaa, Mazin, Momtaaz and Omar Jabir; Maysoon Abd Al-Rahman; Lamees Al-Zubaidy; Sarah and Hajir Basheer; Ruqaya Nazar; Suad Moath; Bashaeer Siddik; Mahir Al-Dabbagh; Mohammed Al-Kattib; Nawar Al-Obaidy; Ienas Younis; Amir and Ali Al-Askery; Dr. Sabah Nayif; Mohammed Abdul-Muttalib; Musab Al-Kaltakchi; Saadoon Al-Sumaidaee; Dr. Jamal Ahmed; Dr. Mohamad Ahmed; and all of my children. I believe that there are many more people who I do appreciate their valuable support, even if did not mention their names.
In addition, my sincere thanks are also given for having access to the "The Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database version 1.0", "IIT Delhi Palmprint Image Database version 1.0" and "Portions of the research in this [thesis] use the CASIA-MS-PalmprintV1 collected by the Chinese Academy of Sciences' Institute of Automation (CASIA)".

Contents

List of Acronyms **xxiv**

1 General overview of biometrics and background for the thesis **1**

1.1 Introduction 1

1.2 Biometrics 2

1.2.1 Biometric Characteristic Types and Specifications 2

1.2.2 Biometric Systems 6

1.2.3 Biometric Fusion Types 8

1.3 The Background for the Thesis 11

1.3.1 Aims and Objectives 11

1.3.2 Thesis Contributions 13

1.3.3 Thesis Outline 16

1.4 Summary 18

2 Finger Characteristics and Relevant Literature Review **19**

2.1 Introduction 19

2.2 Various Finger Characteristics 20

2.2.1 Fingerprints 20

2.2.2 Finger Geometry 20

2.2.3 Finger Veins 22

2.2.4 Finger Outer Knuckle 24

2.2.5 Finger Inner Knuckle 27

2.2.6 Finger Texture 29

2.3 Related Finger Texture Literature Review 31

2.3.1 Segmenting Fingers and Extracting their Textures 31

2.3.2 Prior Work on Feature Extraction 33

2.3.3 Literature Review of Multi-Object Fusion 36

2.3.4 Related Work on Performance Measurements 38
2.4 Summary 40

3 Database Overview **42**
3.1 Introduction 42
3.2 The Databases of the Finger Textures 43
3.2.1 The Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database (Version 1.0) 43
3.2.2 IIT Delhi Touchless Palmprint Database (Version 1.0) 43
3.2.3 CASIA Multi-Spectral Palmprint Image Database (Version 1.0) 54
3.2.4 The Hong Kong Polytechnic University Finger Image Database (Version 1.0) 59
3.3 Performance Measures 61
3.4 Summary 62

4 Finger Segmentation and Extracting the Region of Interest **64**
4.1 Introduction 64
4.2 Main Points of the Fingers 65
4.3 Proposed Finger Segmentation Approaches 67
4.3.1 Robust Finger Segmentation Method 67
4.3.2 Adaptive and Robust Finger Segmentation Method 75
4.4 Extracting the Region of Interest 81
4.5 Results and Comparisons 83
4.6 Summary 88

5 Feature Extraction **90**
5.1 Introduction 90
5.2 Basic Local Binary Pattern 91
5.3 Enhanced Local Binary Patterns 92
5.3.1 Multi-Scale Local Binary Patterns 92
5.3.2 Improved Local Binary Pattern 93
5.3.3 Simplified Local Line Binary Pattern 94
5.3.4 Three-Patch Local Binary Pattern and Four-Patch Local Binary Pattern 95
5.3.5 Local Line Binary Pattern 98

5.3.6 Gradient Directional Pattern 99
5.3.7 Improved Local Binary Pattern Neighbours 101
5.3.8 Local Gradient Coding and Local Gradient Coding-Horizontal Diagonal 102
5.4 Proposed Feature Extraction Approaches 103
5.4.1 Multi-scale Sobel Angles Local Binary Pattern 103
5.4.2 Enhanced Local Line Binary Pattern 107
5.5 Results and Comparisons 109
5.5.1 Evaluating the Suggested MSALBP Feature Extraction 109
5.5.2 Evaluating the ELLBP 112
5.5.3 Texture and Timing Comparisons with Related Work 117
5.6 Summary 124

6 Finger Fusions, Missing Finger Texture Elements and Salvage Approach 126
6.1 Introduction 126
6.2 Artificial Neural Network 127
6.3 Finger Fusions 130
6.3.1 Feature Level Fusion with the Probabilistic Neural Network . 130
6.3.2 Proposed Finger Contribution Fusion Neural Network 133
6.4 Missing FTs and salvaging data approach 136
6.5 Results and Discussions 142
6.5.1 Evaluating the finger fusion methods 142
6.5.2 Evaluating the missing FTs and the salvage approach 144
6.6 Summary 160

7 Conclusions and Future Work 161
7.1 Introduction 161
7.2 Conclusions and Main Findings 162
7.2.1 Descriptions of Employed Databases 162
7.2.2 Finger Segmentation Approaches 162
7.2.3 Feature Extraction Approaches 163
7.2.4 Finger Fusions 164
7.2.5 Missing FT features 165

7.2.6 ROC graphs 167
7.3 Future Work 168

A Receiver Operating Characteristic **170**
A.1 Introduction 170
A.2 Theoretical Concept of the ROC 171
A.3 Generation of ROC for PNN 173
A.4 Generation of ROC for FCFNN 174
A.5 Conclusion of Appendix A 175

References **176**

List of Figures

1.1 Different examples of physiological traits of a human body, where each trait can be used as a physiological biometric characteristic . . . 3

1.2 Various examples of behavioural traits of a personal style, where each trait can be utilized as a behavioural biometric characteristic 4

1.3 The main operations of the enrolment, verification and identification modes. The parameter N represents the total number of the processed biometric enrolment vectors and the parameter Q represents a specific stored vector in the template 7

1.4 The common four biometric fusions of multi-modal biometric systems. The parameter N in all sub-figures represents the number of employed biometric characteristics 10

1.5 The basic hand parts (palm face), these are a single palm and five fingers. The five fingers are a thumb, index, middle, ring and little finger 12

1.6 The main positions of the FTs. Essentially, they can be found in the inner hand surface of the five fingers 12

1.7 The main block diagram of the proposed finger verification scheme; the switches are used simultaneously to change between the training and testing phases 13

2.1 Samples of fingerprint patterns. Each row shows images belonging to a subject and each column shows different acquired images. These images can be found in Set B of Database 2 in [2] 21

2.2 Common finger geometry features of the four fingers, where these are for each single finger: multiple finger widths in assigned locations and finger length [3] 22

2.3 Example of an acquired FV environment [4] 23
2.4 FV image of an index finger 23
2.5 FV image of an index finger shows parts of veins disappeared because of over illumination 24
2.6 FV image of a middle finger 24
2.7 FV image of a middle finger shows parts of veins disappeared because of over illumination 24
2.8 Some ROI images of the FOK 'major' pattern from the PolyUFKP database as shown in [5] 25
2.9 'Major' features of three different individuals [6]; each row represents a 'Major' feature of a participant and each column represents offered samples from the different participants 26
2.10 The locations of the three inner knuckles: upper, middle and lower . . 27
2.11 Samples of middle FIKs as demonstrated in [7] show various patterns of inner middle knuckles 28
2.12 Various patterns that formed the inner surface of a finger: ridges, visible lines and skin wrinkles as given in [8] 29
2.13 The fundamental parts of FTs of a single finger 30

3.1 Example of a hand image from the PolyU3D2D database 44
3.2 Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 0.2cm to the left 44
3.3 Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 1cm to the left 45
3.4 Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 1.75cm to the left 45
3.5 Example of a hand image from the IIT Delhi database 47
3.6 Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 10° to the left 47
3.7 Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 20° to the right 48

3.8 Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 30° to the right 48
3.9 Sample of a hand image from the IIT Delhi database showing a middle finger bent to the back because of restricted space 49
3.10 Sample of a hand image from the IIT Delhi database illustrating a middle finger more significantly bent to the back because of restricted space 49
3.11 Sample of a larger hand image from the IIT Delhi database demonstrating a middle finger bent to the back because of restricted space 50
3.12 Sample of a large hand image from the IIT Delhi database with bent middle, index and ring fingers to the back 50
3.13 Sample of a hand image from the IIT Delhi database with a bent thumb and a rotated hand of approximately 25° to the right direction, where a small amount of texture from the thumb has been lost 51
3.14 Sample of a hand image from the IIT Delhi database illustrating a distorted hand image because of instability during the capturing operation 51
3.15 Example of a hand image from the IIT Delhi database showing a gold ring appearing in the ring finger 52
3.16 Example of a hand image from the IIT Delhi database demonstrating a silver ring in the middle finger 52
3.17 Example of a hand image from the IIT Delhi database with gold and silver rings on the ring and middle fingers 53
3.18 Example of a hand image from the IIT Delhi database showing two gold rings appearing on the ring and middle fingers 53
3.19 A demonstration of the multi-spectral hand image acquisition device for the CASIAMS database as given in [9] 54
3.20 Sample of a right hand image from the CASIAMS database for the wavelength lighting 460nm, where the outer texture of the skin is clarified 56
3.21 Sample of a right hand image from the CASIAMS database for the wavelength lighting 630nm, where the inner veins and outer texture of the skin are shown 56

3.22 Sample of a right hand image from the CASIAMS database for the wavelength lighting 700nm, where the inner veins and outer texture of the skin are shown 57
3.23 Sample of a right hand image from the CASIAMS database for the wavelength lighting 850nm, where the inner veins of the skin are demonstrated 57
3.24 Sample of a right hand image from the CASIAMS database for the wavelength lighting 940nm, where the inner veins of the skin are demonstrated 58
3.25 Sample of a right hand image from the CASIAMS database for the white lighting, where the outer texture of the skin is clarified 58
3.26 Representation of the FV and FT image capturing device [10] 60
3.27 Samples of FT images from the PolyUFI. Each FT image belongs to a subject 60
3.28 Samples of FV images from the PolyUFI. Each FV image belongs to a corresponding subject to FT image in the previous figure 60

4.1 A hand image boundary with the essential points: tips, valleys and symmetric points where $\{T_0, T_1, T_2, T_3$ and $T_4\}$ represent the tips, $\{V_0, V_1, V_2$ and $V_3\}$ represent the valleys and $\{R_0, R_1$ and $R_2\}$ represent the symmetric 66
4.2 Original grayscale hand image 69
4.3 Converted binary image from the grayscale image 70
4.4 Binary image after deleting the small white noise areas 70
4.5 Image complement of the previous binary image 71
4.6 Image complement after the 'majority' morphological filter 71
4.7 Thumb object with the tip and valley points 72
4.8 Four finger objects each object assigned by a specific integer value . . 72
4.9 Four finger objects with the tip points 73
4.10 Background objects assigned by specific colours with the valley points 73
4.11 Original hand image with the tips and valleys points 74
4.12 An example of a hand image with its segmented fingers 74
4.13 Applying a binarization threshold to the red, green and blue channels of a coloured image 77

4.14 Wrong three objects each with an assigned colour 78
4.15 Wrong five objects each with an assigned colour 79
4.16 True four objects represent the four fingers 79
4.17 The anchor point (the valley point between the thumb and index finger) 80
4.18 Sample of a hand image with all tips and valley points 80
4.19 Segmented finger images with their extracted ROIs according to the adaptive inner rectangle method. From the top: the first row is for the thumb; the second row is for the index finger; the third row is for the middle finger; the fourth row is for the ring finger; the fifth row is for the little finger. From the right: the first column is for the segmented finger images; the second column is for the extracted ROIs according to the adaptive inner rectangle method; the third column is for the fixed resize of the ROI images 82

5.1 An example of computing the LBP code for a 3×3 window 92
5.2 Different MSLBP operators, from the left $MSLBP_{8,1}$, $MSLBP_{16,2}$ and $MSLBP_{24,3}$ 93
5.3 An example of computing the ILBP code for a 3×3 window according to [11] 94
5.4 An example of the SLBP operator 95
5.5 The main diagram of the TPLBP operator [12] 96
5.6 The main diagram of the FPLBP operator [12] 96
5.7 An example of the LLBP operator with the vector length of 15 pixels 98
5.8 An example of computing a GDP code 100
5.9 An example of the ILBPN operator 101
5.10 The gradient pixel positions in the LGC kernel [13] 103
5.11 A simple mathematical example of calculating the LGC code 103
5.12 The block diagram of the suggested scheme for the MSALBP feature extraction and the verification operation 104
5.13 The Sobel operator masks: (a) The horizontal edge detection operator and (b) The vertical edge detection operator 104

5.14 Image analysis of the MSALBP operator: Each row represents a ROI of a finger; from the top: thumb, index, middle, ring and little finger respectively. While, the first column is assigned for the original FT images, the second column shows the horizontal edge images, the third shows the vertical edge images and the last row represents the $MSLBP_{8,2}$ of the angle images 105
5.15 A demonstrated example of the suggested ELLBP (N=17) 108
5.16 Example of five ROI finger images and their ELLBPs: the first column in the left represents ROI of each finger and the last column represents the ELLBP image for each finger. While the first row is for the thumb, the second row is for the index finger, the third row is for the middle finger, the fourth row is for the ring finger and the fifth row is for the little finger 109
5.17 The performance of the $MSALBP_{8,1}$ by using various blocking sizes . 110
5.18 The performance of the best MSALBPs by using various blocking size, where the $MSALBP_{8,2}$ has attained the best features for the PolyU3D2D and S460 from the CASIAMS databases and $MSALBP_{16,2}$ has the best features for the IIT Delhi database 112
5.19 The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the PolyU3D2D database 113
5.20 The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the IIT Delhi database 113
5.21 The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the CASIAMS (S460) database 114
5.22 The performance of the ELLBP (N=17) by using various blocking sizes 115
5.23 The performance of the ELLBP (N=7) for the CASIAMS (S460) by using various blocking size 117
5.24 ROC curves of various LBP types which have been used in this study for the PolyU3D2D database (The Y-axis scale is reduced to make the figure more clearer) 119
5.25 ROC curves of various LBP types which have been used in this study for the IIT Delhi database 120

5.26 ROC curves of various LBP types which have been used in this study for the CASIAMS (S460) database 120

6.1 A biological neural network and a simple neural network. The biological neural network is shown at the left [14] and a simple neural network is demonstrated at the right 128
6.2 A single layer ANN consisting of an input layer, an output layer and a single layer of connections 128
6.3 A multiple layer ANN which consists of an input layer, a hidden layer, an output layer and two layers of connections 128
6.4 A fully-connected multiple layer ANN which consists of an input layer, a hidden layer, an output layer and three layers of connections 129
6.5 The main diagram of the suggested FLFPNN verification system; the switches should be used together to swap between the training and the testing phases 130
6.6 The general form of the PNN with the concatenated five finger feature input vector of the five fingers. It includes input, pattern, summation and decision layers 132
6.7 The proposed FCFNN method including the contribution layer 134
6.8 Step one of a salvage example clarifying a simple training of a PNN . 138
6.9 Step two of a salvage example clarifying a testing phase, the EER has attained 0% 139
6.10 Step three of a salvage example clarifying a testing phase with missing half number of elements, the EER has been increased to 33.33% . . . 140
6.11 Step four of a salvage example clarifying a testing phase with the salvage process, the EER has been decreased to 0% 141
6.12 The ROC curves of the MSALBP(P=8,R=2) feature extraction with the FLFPNN and for the PolyU3D2D database (The Y-axis scale is reduced to make the figure clearer) 145
6.13 The ROC curves of the ELLBP(N=17) feature extraction with the FLFPNN and for the PolyU3D2D database (The Y-axis scale is reduced to make the figure clearer) 146
6.14 The ROC curves of the MSALBP(P=16,R=2) feature extraction with the FLFPNN and for the IIT Delhi database 146

6.15 The ROC curves of the ELLBP(N=17) feature extraction with the FLFPNN and for the IIT Delhi database 147
6.16 The ROC curves of the MSALBP(P=8,R=2) feature extraction with the FLFPNN and for the CASIAMS (S460) database 147
6.17 The ROC curves of the ELLBP(N=7) feature extraction with the FLFPNN and for the CASIAMS (S460) database 148
6.18 Hand images in the PolyU3D2D database show the illumination distribution and the thumb appears to be brighter than the remaining fingers because of the acquisition lightning environment . . 151

List of Tables

3.1 The wavelengths and frequencies of the visible spectrum colours . . . 59

4.1 Comparison summary between the suggested segmentation methods and related work . . . 85
4.2 Comparison summary between the exploited ROIs extraction method and prior studies . . . 86
4.3 The EER results for four fingers without inner knuckles, four fingers with inner knuckles and five fingers with inner knuckles . . . 88

5.1 Sobel amplitude values versus Sobel angle directions . . . 110
5.2 The verification performance of the different MSALBP parameters . . . 111
5.3 The EER results for ELLBP lengths of the all databases by using the PNN classifier . . . 116
5.4 The EER results for ELLBP lengths of the CASIAMS (S460) Database by using the FCFNN classifier . . . 117
5.5 Comparisons between the verification performance of different LBP types for the FTs of the three databases . . . 118
5.6 Averaged timing comparison between the different LBP operators for a single finger . . . 123

6.1 Verification performance comparisons between the FLFPNN and the FCFNN using best feature extraction achievements . . . 143
6.2 EERs before and after the suggested salvaging method for the MSALBP feature extraction with the FLFPNN . . . 149
6.3 EERs before and after the suggested salvaging method for the ELLBP feature extraction with the FLFPNN . . . 150
6.4 EERs before and after the suggested salvaging method for the MSALBP feature extraction with the FCFNN . . . 153

6.5 EERs before and after the suggested salvaging method for the ELLBP feature extraction with the FCFNN 154
6.6 Removing a finger or fingers from the FCFNN using the MSALBP feature extraction 156
6.7 Removing a finger or fingers from the FCFNN using the ELLBP feature extraction 157

A.1 The confusion matrix 171
A.2 Summary of related approaches for generating the ROC graph from a PNN 172

List of Acronyms

ANN	Artificial Neural Network
ARFS	Adaptive and Robust Finger Segmentation
ATM	Automated Teller Machine
CASIAMS	CASIA Multi-Spectral
CCD	Charge Coupled Device
CMC	Cumulative Match Curve
CompCode	Competitive Coding
COV	Coefficient of Variance
DET	Detection Error Tradeoff
DNA	DeoxyriboNucleic Acid
EER	Equal Error Rate
ELLBP	Enhanced Local Line Binary Patterns
FAR	False Acceptance Rate
FCFNN	Finger Contribution Fusion Neural Network
FG	Finger Geometry
FIK	Finger Inner Knuckle
FLFPNN	Feature Level Fusion with a Probabilistic Neural Network
FOK	Finger Outer Knuckle
FPLBP	Four-Patch Local Binary Pattern

FRR	False Rejection Rate
FT	Finger Texture
FV	Finger Veins
GDP	Gradient Directional Pattern
$GRNN$	General Regressive Neural Network
HD	Hamming Distance
IFE	Image Feature Enhancement
IIT	Indian Institute of Technology
$IITDFK$	Indian Institute of Technology Delhi Finger Knuckle
$ILBP$	Improved Local Binary Pattern
$ILBPN$	Improved Local Binary Pattern Neighbours
$JPEG$	Joint Photographic Experts Group
LBP	Local Binary Pattern
LED	Light-Emitting Diode
LGC	Local Gradient Coding
$LGC-HD$	Local Gradient Coding-Horizontal Diagonal
$LLBP$	Local Line Binary Pattern
MLP	Multi-Layer Perceptron
$MSALBP$	Multi-scale Sobel Angles Local Binary Pattern
$MSLBP$	Multi-Scale Local Binary Pattern
NIR	Near-InfraRed
PDF	Probability Density Function
PNN	Probabilistic Neural Network
$PolyU3D2D$	Polytechnic University Contact-free 3D/2D

$PolyU3D2DV2$	Polytechnic University Contact-free 3D/2D Hand Images Database (Version 2.0)
$PolyUCFKI$	Polytechnic University Contactless Finger Knuckle Images
$PolyUFI$	Polytechnic University Finger Image
$PolyULRF$	Polytechnic University Low Resolution Fingerprint
$PPNN$	Parzen PNN
RBF	Radial Basis Function
$RBFNN$	Radial Basis Function Neural Network
RFS	Robust Finger Segmentation
ROC	Receiver Operating Characteristic
ROI	Region of Interest
$S460$	Spectral 460nm
$SLBP$	Simplified Local Binary Pattern
SVM	Support Vector Machine
$TPLBP$	Three-Patch Local Binary Pattern
TPR	True Positive Rate

Chapter 1

General overview of biometrics and background for the thesis

1.1 Introduction

The term 'biometric' is currently widely used to indicate a way of recognizing people. It has been explored for many years in different applications. Examples of these applications are security systems, recognizing people and forensic investigations. Many biometric characteristics within the human body have been separately investigated such as the iris [15, 16], sclera [17, 18], face [19, 20], palmprint [21, 22], voice [23, 24] and Finger Texture (FT) [25–27]. Furthermore, various studies have also exploited more than one biometric characteristic such as [28–31]. In this chapter, the following points will be considered:

- General overview of the fundamentals of biometrics, and
- Background of the thesis.

The fundamentals of biometrics will firstly be described. Two basic types physiological and behavioural will be introduced, with brief illustrations of common characteristic examples. The concept of how to consider a personal trait as a biometric characteristic will be explained by highlighting the biometric specifications. Then, the three modes of a biometric system will be stated, these are enrolment mode, verification mode and identification mode. Finally, the essential perceptions of the four biometric fusion strategies will be demonstrated. On the other hand, the second part of this chapter will focus on the targets of this

thesis. Thus, the FT subject and its utilities will be illustrated. The aims with the objectives will be given. The contributions will be highlighted. Finally, the organization of the thesis will be outlined.
The rest of this chapter will include: Section 2 which presents the basic concepts of biometrics, Section 3 which illustrates the targets and configurations of the thesis, and Section 4 will summarize the chapter.

1.2 Biometrics

1.2.1 Biometric Characteristic Types and Specifications

Biometric recognition can be considered as one of the most important parts of a large number of authentication systems and technologies to allow secure access. Examples of some technologies which can employ a biometric characteristic to authenticate the automatic access are mobile phones, laptops, private computers and Automated Teller Machines (ATMs) [32]. Efficient biometric systems that are using more than one biometric characteristic can be found specifically in some high security buildings. Without biometric recognition such products and buildings can be easily attacked by unlicensed or unauthorized people.
In general, biometric characteristics can be categorized into two types as follows:

1. Physiological biometrics: this type of biometric refers to the physiological traits within the human body such as fingerprints, iris print and palm print. Fig. 1.1 shows different examples of physiological biometric traits.

2. Behavioural biometrics: this type of biometric points to the behavioural traits of the individual style such as signature, keystroke and voice. Fig. 1.2 demonstrates various examples of behavioural biometric traits.

The physiological biometrics are usually more accurate and reliable, whereas the behavioural biometrics may be affected by the emotional feelings such as sickness or tension [33]. Brief descriptions of common physiological and behavioural traits are illustrated as follows:

- Fingerprint: this is a well known biometric, it refers to the unique patterns of the skin surface at the top of each finger [30, 34–36].

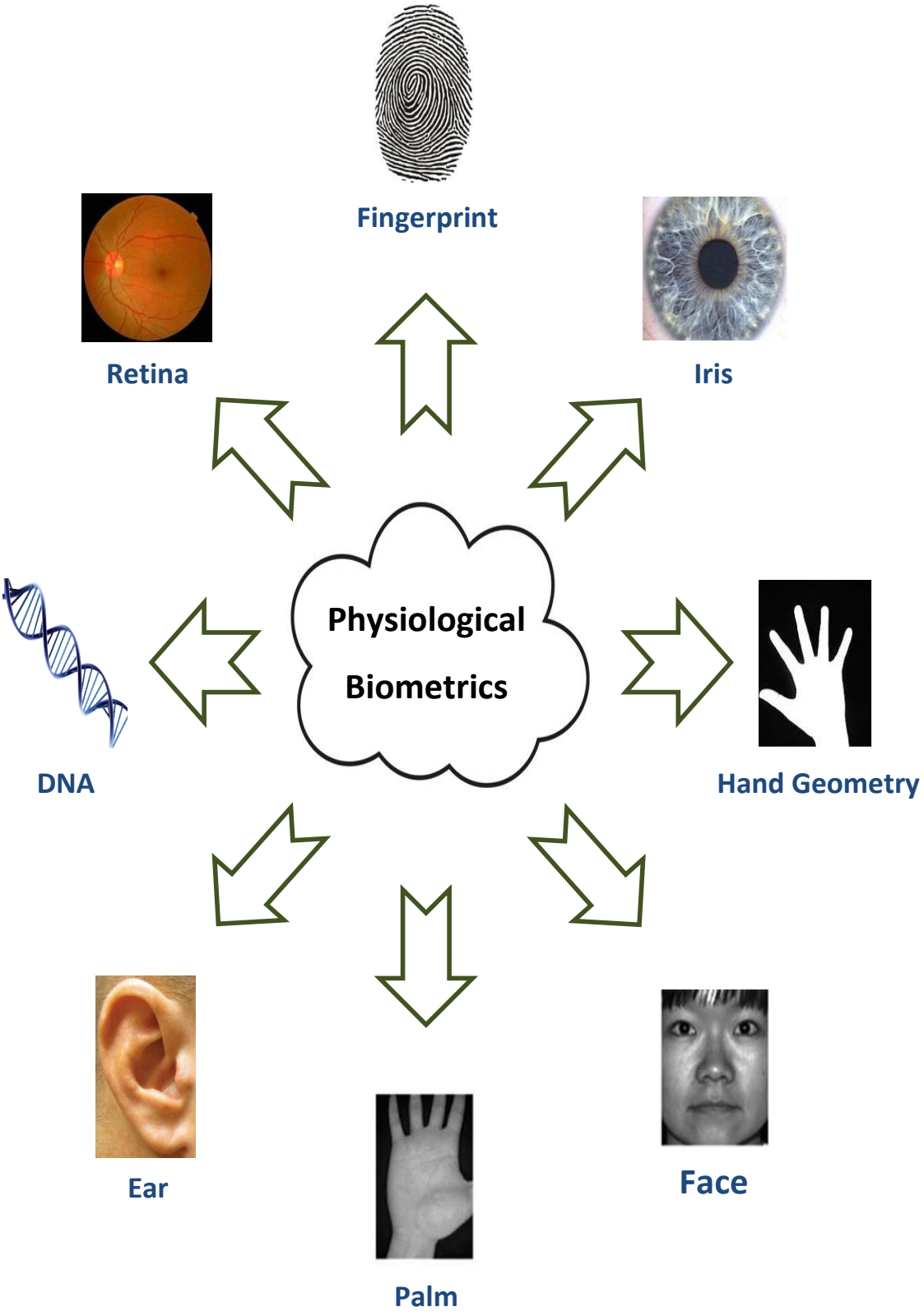


Figure 1.1: Different examples of physiological traits of a human body, where each trait can be used as a physiological biometric characteristic

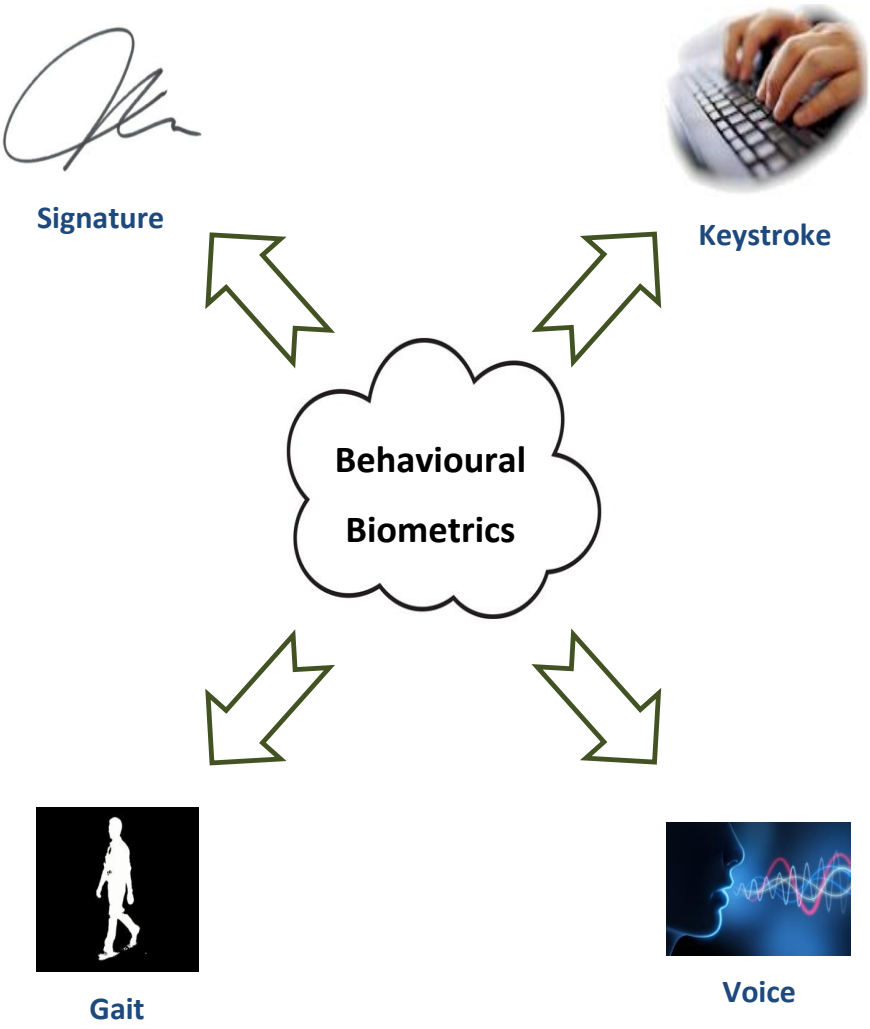


Figure 1.2: Various examples of behavioural traits of a personal style, where each trait can be utilized as a behavioural biometric characteristic

- Iris: the iris texture is formed from the muscle structures of the iris, it can be found in the coloured area of an eye between the pupil and sclera, and it was widely used for personal recognition [15, 37, 38].
- Hand Geometry: the hand geometry mainly considers the general specifications of the hand shape such as the lengths of fingers and the widths of fingers and the palm [39, 40].
- Face: this trait has been used in different fields such as face recognition [41], face expression recognition [42] and face detection [43], because it includes various objects and rich textures.
- Palm: the palm surface of the hand consists of a reliable structure of features,

these features are ridges, wrinkles and lines [44, 45], so, any of these features can be exploited in the case of personal recognition.

- Ear: the shape of the ear is employed to verify people in secure telephone systems, where ears are usually used for the phone call purposes [46,47]. More information regarding ear recognition can be found in [48].

- DeoxyriboNucleic Acid (DNA): this trait has an interesting sequence of beneficial codes, it has been utilized efficiently in forensic investigations to recognize suspected people [32, 32].

- Retina: retina is located inside the eyeball on the back surface, it has exclusive patterns of vein structures which can be employed in terms of personal recognition [33, 49].

- Signature: this trait depends on the human style of providing a certain signature, where different features can be considered such as the graphic shape of the signature and the distribution density of its handwriting pattern [50, 51].

- Keystroke: the tone of the keystroke is different among individuals, thus, this has been considered as a biometric characteristic too [52, 53].

- Voice: this biometric trait has been analysed and adopted in the case of speaker recognition by many publications such as [23, 24, 54].

- Gait: each person has a certain gait, so, the manner of walking could be exploited in identifying or verifying people [55, 56].

A number of specifications are required for each physiological or behavioural trait in order to be considered as a biometric characteristic. These specifications are highlighted as follows [32]:

- Stability: the biometric patterns have to be stable and permanent.

- Uniqueness: the features of the trait must be unique and vary between any two individuals.

- Collectability: this means that the collected biometric features should be valuable and measurable.

- Popularity: the trait should be popular or non-exclusive for certain individual(s).

In addition, the biometric systems are highly preferable to have the following factors [32]:

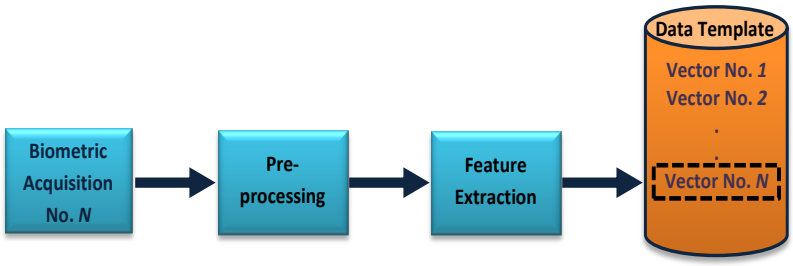
- Suitability: this refers to the acceptability as a user-friendly biometric system.
- Implementation: the biometric system has to be highly applicable and reliable.
- Circumvention: the biometric system should be difficult to avoid by fraudsters.

Therefore, it is necessary to consider the previous points to establish a robust biometric system.

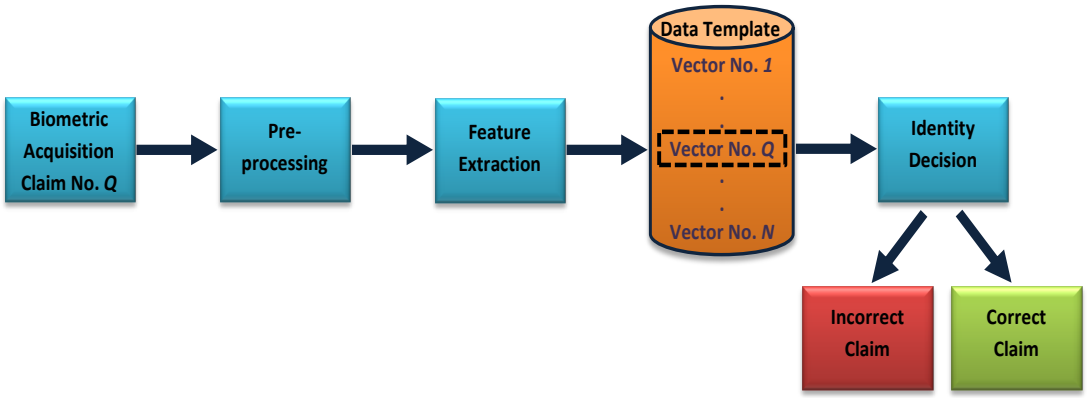
1.2.2 Biometric Systems

Basically, any biometric system consists of a sequence of operations starting from acquiring a biometric characteristic, pre-processing operations such as segmentation, extracting features and generating output decision according to the identity matching method [57].
A biometric system is generally recognized as an identification system or a verification system. In fact, a biometric system can be designed to work in one of three modes as will be detailed below:

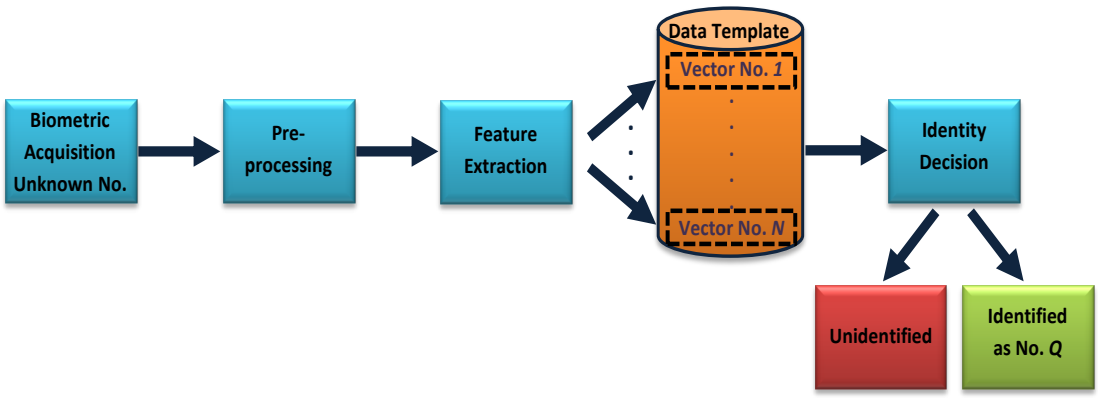
- Enrolment mode: this is an initial step of any biometric system, where a template is created by storing features of the enrolled biometric characteristics. The enrolment mode is working as one-after-one policy. That is, each input has to be treated separately and independently starting from the capturing or scanning step and ending with distinctly saving its extracted features in the template. See Fig. 1.3a.
- Verification mode: in this mode a user claims his identity. So, the similar operations of biometric acquiring, pre-processing and feature extraction are implemented. Then, the resulting feature vector will be matched with the same claimed identity vector. This module is known as a one-to-one matching. Finally, the identity decision is to confirm or reject the claim. Fig. 1.3b shows the general operations of the verification mode.



(a) Enrolment mode, there are no comparisons in this mode as it just to store the features of the current enrolment input inside the template



(b) Verification mode, in this mode a comparison is achieved by matching the features of the current input with the features of a claimed stored vector inside the template



(c) Identification mode, in this mode a comparison is achieved by matching the feature vector of the current input with all of the stored feature vectors inside the template

Figure 1.3: The main operations of the enrolment, verification and identification modes. The parameter N represents the total number of the processed biometric enrolment vectors and the parameter Q represents a specific stored vector in the template

- Identification mode: like the prior operations of the verification and enrolment modes, the identification will extract the feature of the input biometric trait. However, a one-to-many matching will be implemented between the extracted feature vector and all of the stored vectors in the template. In this mode, the user cannot provide his identity. A decision is established by assigning the identity of the user or rejecting his/her membership to the biometric system. The main operations of the identification mode are demonstrated in Fig. 1.3c.

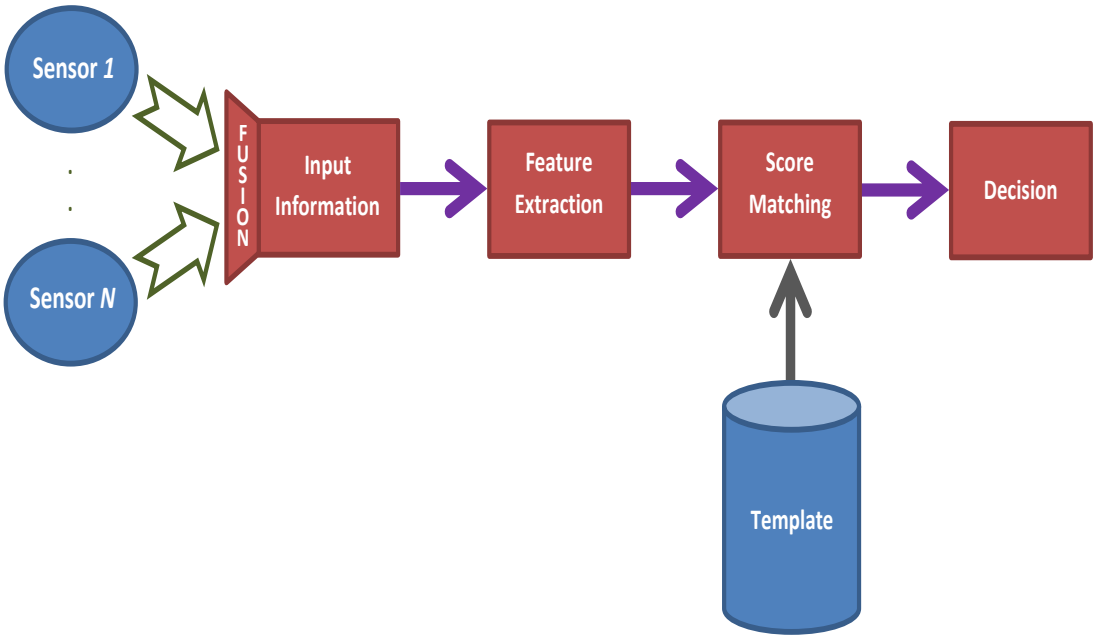
1.2.3 Biometric Fusion Types

To enhance the error performance of a single-modal biometric system, which employs only one biometric characteristic, a fusion between multiple biometric characteristics is usually applied [58–60]. The fusion between multiple biometric characteristics in one biometric system is called a multi-modal biometric system [61]. Generally, there are four levels of fusion: sensor level fusion, feature level fusion, score level fusion and decision level fusion [62]. Fig. 1.4 depicts the topography or the architecture of the four fusion levels and they can be described as follows:

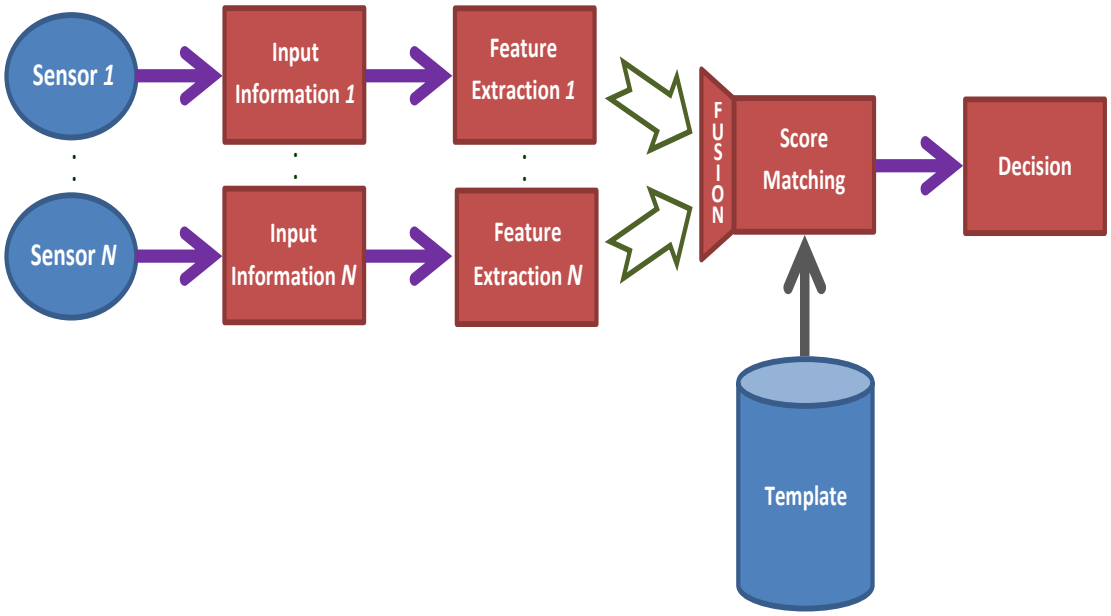
1. Sensor level fusion: this is the first level of fusion or combination as it performs at the sensor level. With this type of fusion more than one sensor should be provided to a biometric system. The basic idea here is that each sensor can collect certain features, thus, by combining the acquired biometric data more information will be analysed in the biometric system. It can be seen from Fig. 1.4a that the sensors will capture various biometric data and these data will be fused before the input information stage. Then, the remaining operations will be carried out after the combination step.

2. Feature level fusion: this is the second level of fusion. It can be implemented after the feature extraction step, where the input information can be analysed by using different feature extraction methods. Consequently, the resulting features are combined after applying several treatments such as normalizing the information and reducing the overall size. In this case more extracted features will be considered in the biometric system. See Fig. 1.4b.

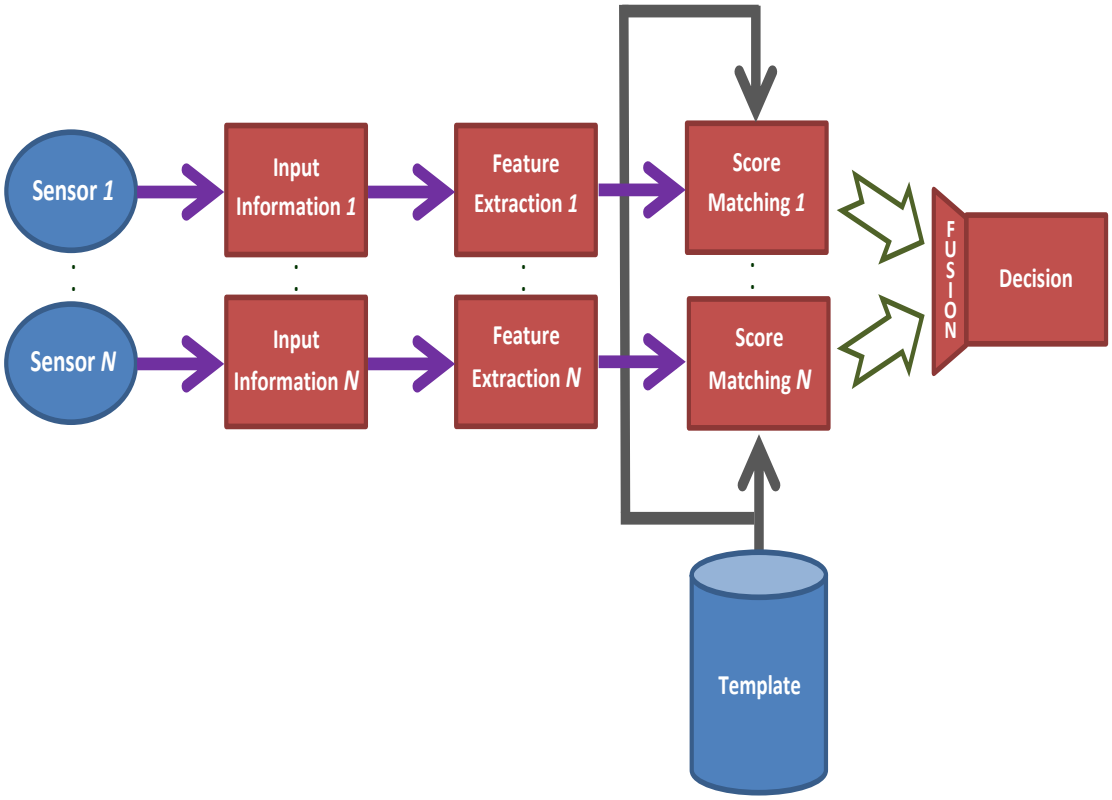
3. Score level fusion: this is the third combination level. It refers to the fusion between the scores of the employed matchers, where various scores can result



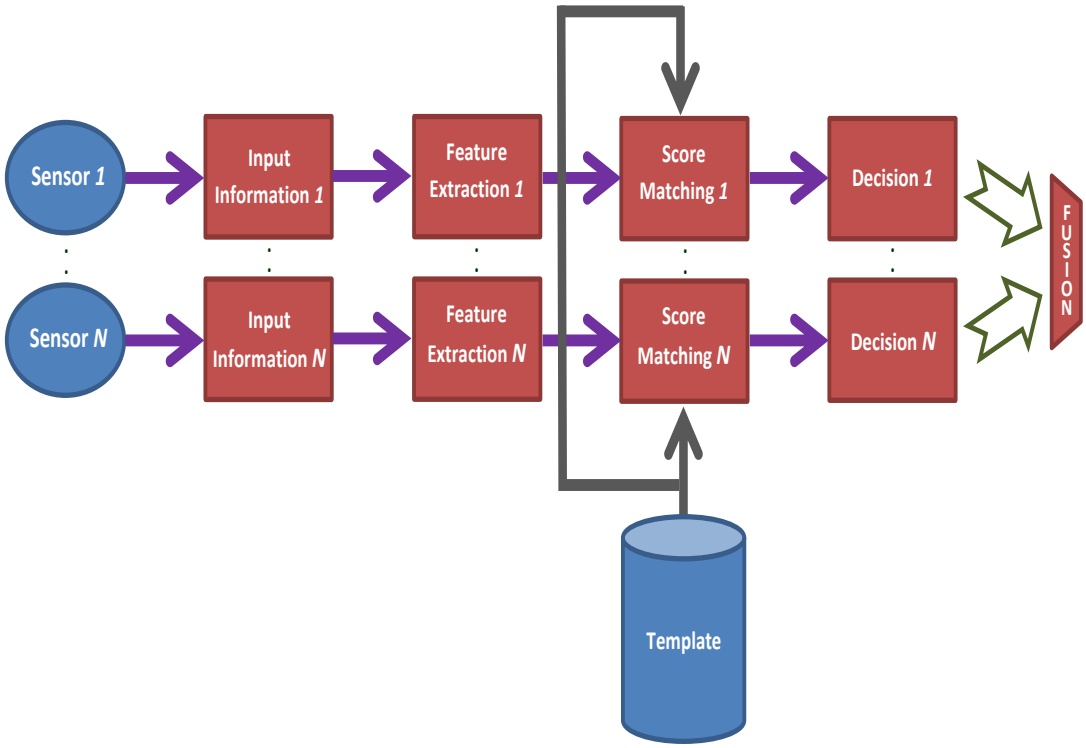
(a) Sensor level fusion by combining the outcomes of two or more sensors, each sensor can capture a different biometric characteristic. However, same operations are used to analyse the input information after the fusion



(b) Feature level fusion by combining the outcomes of two or more featured vectors, various biometric characteristics can be processed by using different methods before the fusion; whereas, the same remaining operations after the fusion are performed



(c) Score level fusion by combining the outcomes of two or more matcher scores. Again, different characteristics and different analysing methods can be used before the fusion, whilst the same decision operation is implemented after the fusion



(d) Decision level fusion by combining the outcomes of two or more decision outputs. This is the last fusion method after various operations, where the outputs can be obtained immediately after the fusion

Figure 1.4: The common four biometric fusions of multi-modal biometric systems. The parameter N in all sub-figures represents the number of employed biometric characteristics

from multiple matchers. Therefore, combining these scores can improve their overall effects on the decision operation. A general topography of a score fusion for a multi-modal biometric system is given in Fig. 1.4c.

4. Decision level fusion: this is the last fusion level and it is also the last operation of a biometric system. The output of the biometric system here is the final indication of the combination between all of the available decisions. So, there is no operation or process after this step. A general diagram of a decision fusion is demonstrated in Fig. 1.4d.

It is worth mentioning that any fusion level can be used for a single-modal biometric system by employing only one characteristic such as in [5, 63]. Nevertheless, they are usually used in a multi-modal biometric system.

1.3 The Background for the Thesis

1.3.1 Aims and Objectives

Inner finger surface texture has drawn considerable attention over about the last 10 years. It has similar patterns as in the palm. These patterns are known as FT, and they mainly include wrinkles and principal lines. It has been considered that the FTs are reliable and unique between individuals or even between identical twins [64]. Furthermore, there are many advantages of using the FTs. For example, they have rich information, they are resistant to emotional feelings, and their patterns are stable and reliable [64]. They are distributed on the inner surface of the five objects or fingers. So, they can be found on the inner surface of the thumb, index, middle, ring and little. Figs. 1.5 and 1.6 show the basic parts of a hand and the main locations of the FTs, respectively.

A biometric verification scheme based on the FTs is proposed. Fig. 6.5 shows the general block diagram of the suggested scheme. It consists of multiple operations and two phases: training and testing. Several operations are related to the similar functions in both phases. Whilst, the switches are designed to be used together to swap between the phases. The verification operations are started by entering a hand image and the learning phase will be started first. Then, the finger segmentation will be carried out. The Region of Interests (ROIs) of the five finger

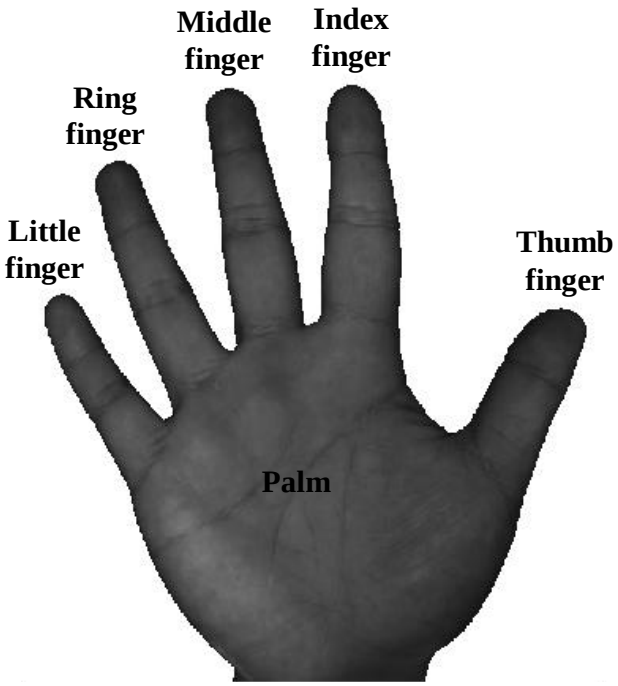


Figure 1.5: The basic hand parts (palm face), these are a single palm and five fingers. The five fingers are a thumb, index, middle, ring and little finger

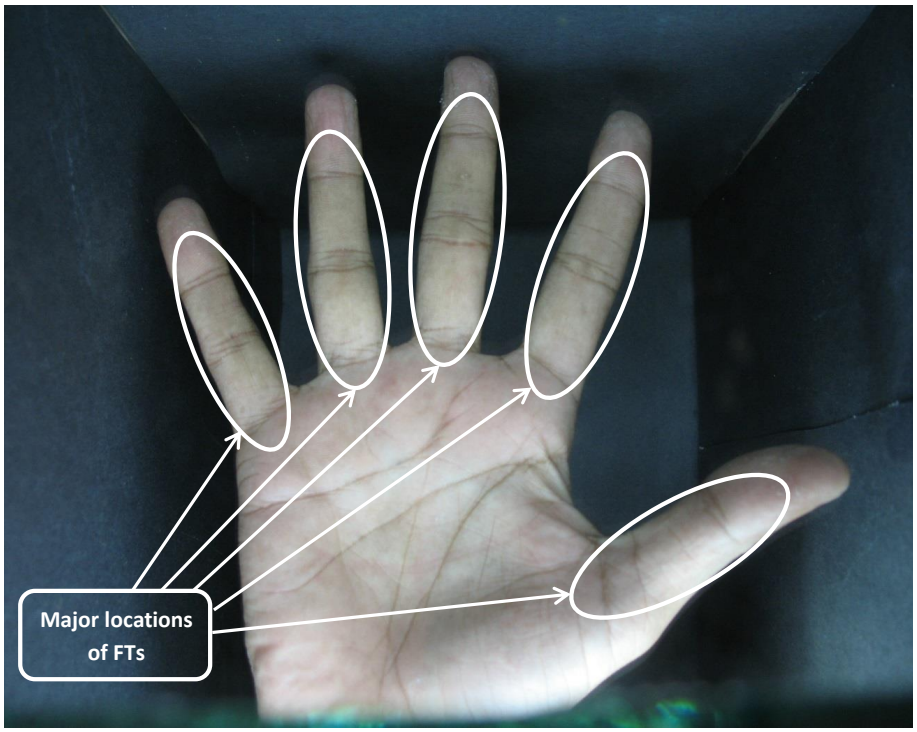


Figure 1.6: The main positions of the FTs. Essentially, they can be found in the inner hand surface of the five fingers

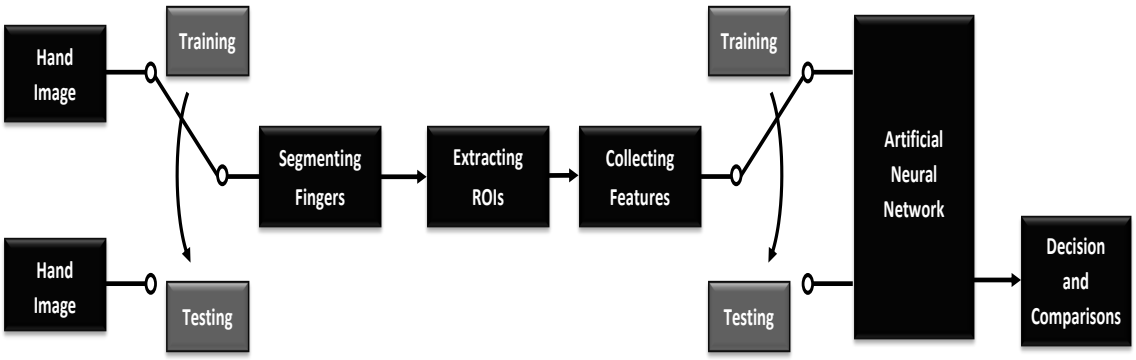


Figure 1.7: The main block diagram of the proposed finger verification scheme; the switches are used simultaneously to change between the training and testing phases

images will be extracted. Hereafter, the feature extractions will be performed to each ROI finger image. Finally, an Artificial Neural Network (ANN) technique is employed in the case of verification.

During the testing phase, all the above operations are repeated, but for new hand images. Also, by taking into account using the trained weights in the ANN. The last step is to obtain the verification decision and comparing different processing approaches.

The aim of this study is to work intensively with the FTs and exploit hand fingers in the task of human verification. Furthermore, this work will contribute to this growing area of research by addressing the previous drawbacks from the prior work such as [25, 58]. In addition to proposing beneficial approaches which can enhance the verification performance. Various comparisons are performed to demonstrate the ability and efficiency of the proposed methods. Supporting results are provided for three databases in order to evaluate the suggested approaches. These databases are Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66], Spectral 460nm (S460) from the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9] and The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67] for the resolutions of high, low and very low, respectively.

1.3.2 Thesis Contributions

There are several important areas where this study makes original contributions:

1. Finger Segmentation:

   - A Robust Finger Segmentation (RFS) method to collect the five finger images from a hand image is established. In this approach, each finger is considered as an object by adopting multiple image processing steps. Furthermore, the RFS is suggested to include more FT features to the segmented fingers by containing the information of the lower knuckles as this could reduce verification error rates.
   - An Adaptive and Robust Finger Segmentation (ARFS) method is presented. This approach is suggested to be adapted to different hand alignments such as rotations and translations. Useful strategies have been proposed to segment the five finger images as objects from a free positioning hand image such as using scanning line, adaptive threshold and adaptive rotation. More FTs are also suggested to be contained in the segmented finger images.

2. Feature Extraction:

   - A new descriptor called the Multi-scale Sobel Angles Local Binary Pattern (MSALBP) is designed. This operator has been created by utilizing the Sobel vertical and horizontal edge angles of the FTs. Subsequently, a Multi-Scale Local Binary Pattern (MSLBP) has been applied to the resulting image. Instead of using the histogram features a simple and efficient statistical calculation is described to collect the variances of the extracted features. It is worth mentioning that each stage of this descriptor has been evaluated.
   - An enhanced feature extraction method is proposed called Enhanced Local Line Binary Pattern (ELLBP). The basis of this approach is to analyse the vertical and horizontal patterns, which are the main patterns in the FT, according to the Local Line Binary Pattern (LLBP) calculations. Consequently, a combination between the vertical and horizontal codes are computed according to the weighted summation rule. Likewise for the MSALBP, statistical calculations are exploited to describe the feature vector. Every stage in this method has also been evaluated.

3. Finger Fusions:

    - A beneficial verification structure is designed and implemented based on the Feature Level Fusion and the Probabilistic Neural Network (FLFPNN). To illustrate, the five finger images are segmented, their ROIs are separately extracted, the features were collected for each ROI and a feature level fusion is implemented. Subsequently, a Probabilistic Neural Network (PNN) is used to process the fusion features.
    - An innovative multi-object fusion method named the Finger Contribution Fusion Neural Network (FCFNN) is proposed as a score level fusion. The PNN is enhanced here to obtain the finger contribution scores by establishing an additional hidden layer called a contribution layer. After that, a summation fusion is calculated to combine these scores together. Finally, the output decision are generated depending on the resulting fusion.

4. Missing FT Elements and Salvage Approach:

    - Verification performance is evaluated for missing FT elements (a small or a big part of a finger, or even a full finger). So, this study is going to answer this question: if a part or full finger is accidentally removed, how this will affect the verification rate? as to the best of the attained knowledge no publication has explored this issue.
    - A novel approach is described to enhance the verification rates in the case of missing a part or full FT by salvaging features embedded in the trained PNN. As due to the fact that the values of the training vectors will be stored in the PNN as weights, this inspired the idea of salvaging the missing feature values.

5. Receiver Operating Characteristic:

    - The aim of this contribution is to present a novel approach to generate the Receiver Operating Characteristic (ROC) graph from a multi-class PNN. This is accomplished by obtaining score values, which are necessary to produce the ROC curve, from the PNN. Hereafter, the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are calculated for each

class. Then, the relationship between the FAR and (1-FRR) is applied in the ROC graph after averaging all of their values.

- A new approach to establish the ROC curve from a FCFNN is presented. The previous contribution is developed to be utilized in this neural network. Again, the most important factors to be collected from the FCFNN are the score values. Subsequently, the FAR and FRR are computed for each class. Finally, the ROC graph is obtained from the relationship between the average values of the FAR and the average values of the (1-FRR).

Supporting results and comparisons are provided for three databases in order to evaluate the suggested approaches. In addition to a large number of experiments, which have been applied.

1.3.3 Thesis Outline

This thesis is organized as follows:

Chapter 1: introduces biometric recognition and provides a general overview of the main two biometric types (the physiological and behavioural biometrics), the specifications of the biometric characteristics and systems, the essential biometric system types (the verification and identification modes) and the biometric fusion types (sensor level fusion, feature level fusion, score level fusion and decision level fusion). Moreover, the aims, objectives, contributions and organization of the thesis are highlighted.

Chapter 2: presents various characteristics related to the finger such as fingerprint, Finger Geometry (FG), Finger Veins (FV), Finger Outer Knuckle (FOK) and Finger Inner Knuckle (FIK). Then, a significant consideration is provided for the FT characteristic. After that, a related literature review is detailed for different stages of the FT biometric system (finger segmentation and ROI extraction, feature extraction, multi-object fusion and performance measurements).

Chapter 3: describes the attributes of the databases that offer the FT patterns. These databases are the PolyU3D2D Hand Images (Version 1.0), the Hong Kong Polytechnic University Contactfree 3D/2D (Version 2.0) (PolyU3D2DV2), the IIT Delhi Palmprint Images (Version 1.0), the CASIAMS (S460) Palmprint images (Version 1.0) [9], the Hong Kong Polytechnic University Finger Image (PolyUFI)

(Version 1.0) and the Hong Kong Polytechnic University Low Resolution Fingerprint (PolyULRF) (Version 1.0). Later, the performance measures are justified.

Chapter 4: demonstrates the finger segmentation approaches. First of all, the main points of the fingers in a hand image are specified. Hereafter, the two segmentation approaches are described. These are the RFS and the ARFS methods. Consequently, extracting the ROIs from each finger image is illustrated. Supported results and comparisons are discussed.

Chapter 5: explains the feature extraction contributions. Firstly, a theoretical part is given for the basic Local Binary Pattern (LBP) and some of its enhancement methods or updated versions such as the MSLBP, Improved Local Binary Pattern (ILBP), Simplified Local Binary Pattern (SLBP), Three-Patch Local Binary Pattern (TPLBP), Four-Patch Local Binary Pattern (FPLBP), LLBP, Gradient Directional Pattern (GDP), Improved Local Binary Pattern Neighbours (ILBPN), Local Gradient Coding (LGC) and Local Gradient Coding-Horizontal Diagonal (LGC-HD). Subsequently, the two suggested feature extraction approaches are detailed. These are the MSALBP and the ELLBP. Evaluations are applied for each approach. Then, extensive comparisons are recorded with updated LBP versions approaches in the case of verification performance.

Chapter 6: clarifies the finger fusions, Missing FT Elements and Salvage Approach. The principles of the ANNs are given. After that, the two suggested finger fusion methods are explained, which are the FLFPNN and FCFNN. Then, the missing FT parts are suggested and salvaging the missing information approach is proposed. The results are discussed for evaluating the two finger fusion methods, examining the missing FT features and applying the salvage approach.

Chapter 7: concludes the thesis by briefly clarifying the main outcomes of this study. Subsequently, this chapter provides recommended future work, which is related to this thesis.

Appendix A: describes generating the ROC curve from a PNN and from the FCFNN. The challenge of this approach is firstly highlighted according to prior studies. The essential parameters of the ROC graph are presented. How to produce the ROC graph from the PNN is explained. Then, how to establish the ROC graph from the FCFNN is illustrated.

Moreover, almost each chapter has an introduction section and a summary section.

These sections are to review the context of that chapter along with their main considerations.

1.4 Summary

As a summary of this chapter, two main parts were included. These were the general concepts of biometrics and the background of this thesis. Relating to the first part, biometric recognition was defined. The main types of biometric characteristics were illustrated, where there are two essential types physiological and behavioural. Brief descriptions of common characteristics were given. The major specifications that should be provided in any biometric characteristic and biometric system were denoted. Furthermore, the three modes that a biometric system can be assigned to any one of them were demonstrated as enrolment, verification and identification. Then, the common fusion styles between multiple biometric characteristics were highlighted as sensor level fusion, feature level fusion, score level fusion and decision level fusion.

On the other hand, the subject of this thesis was assigned for the FT biometric characteristic. So, some significant attributes of this phenomenon were revealed. The aims of the thesis were also illustrated in this chapter. The contributions were clarified for different stages of the FT verification system. Furthermore, the overall organization of the thesis was reported.

In the next chapter, various biometric characteristics which are available in any finger will be described. However, more concentrations will be provided for the FTs. The related literature review of each part of a FT verification system will be provided.

Chapter 2

Finger Characteristics and Relevant Literature Review

2.1 Introduction

Rich characteristics can be observed in each single finger. First of all, they hold the most famous biometric which is the fingerprint. In addition, other attractive biometrics embedded in a finger are the Finger Geometry (FG), Finger Veins (FV), Finger Outer Knuckle (FOK), Finger Inner Knuckle (FIK) and Finger Texture (FT). An overview of these different finger pattern types will be presented. Then, more focus will be given to the FT, as this is the subject of this thesis, and relevant literature review will be provided. So, in this chapter:

- Descriptions of common finger biometrics or characteristics will be demonstrated, and
- Critical analysis of the relevant FT literature review will be given for each step of the personal recognition system.

Each one of the finger characteristics has unique features. Their pattern structures remain stable throughout a human's life. Also, one of the most interesting attributes in any finger characteristic is that its features are different not just between the individuals, in fact among the fingers too. On the other hand, each type requires special considerations in terms of acquiring the biometric image pattern and suggested verification/identification procedure.
In the case of the FT literature review, few works have utilized the FT as a

biometric type. The idea of using this important biometric only appeared approximately one decade ago. Because an FT recognition system is generally compromised within the following steps: finger segmentation and FT collection; feature extraction; matching classifier (multi-object fusion could be used in this stage) and recognition performance, related work will be reviewed for each step.
The rest of this chapter is organized as follows: Section 2 highlights and illustrates common finger characteristic types. Section 3 considers the relevant literature review of FT recognition systems and Section 4 summarizes this chapter.

2.2 Various Finger Characteristics

2.2.1 Fingerprints

Fingerprints have been studied for many years and can be considered as the first effective physiological biometric. They have been employed in many biometric fields such as classification [68], verification [69], identification [70] and multi-modal recognition [71]. Fingerprints are constructed from different pattern forms. These patterns consist of beneficial features such as valleys/ridges, core-points and minutiae [72]. Fig. 2.1 demonstrates various fingerprint images, where the fingerprints that belong to the same subject have similar patterns and the fingerprints that belong to the different subjects have different patterns.
The most important advantage of this biometric compared with other finger patterns is that its traits can be collected without requiring a certain finger to be presented. This facilitates the forensic investigations of crimes [73–78].
However, biometric systems based upon fingerprints have obstacles. For example, it has been reported that recognition rates of fingerprints are generally reduced when the individual becomes older [35]. Furthermore, it has been noticed that fingerprint patterns may vanish for elderly people especially those suffering from diabetes [79]. Therefore, this will cause erroneous results in biometric systems.

2.2.2 Finger Geometry

The geometry of fingers is an important part of hand geometry as described in [80–82]. Several recent studies considered FG alone as a type of biometric characteristic such as in [83–85]. Generally, to extract the geometry feature of a single finger

Figure 2.1: Samples of fingerprint patterns. Each row shows images belonging to a subject and each column shows different acquired images. These images can be found in Set B of Database 2 in [2]

multiple widths in determined locations along with the finger length are utilized in verification/identification systems. These common features of FGs are demonstrated in Fig. 2.2.

The following issues should be considered with the FG characteristic: an appropriate binarization threshold for a hand image is required [86], pegs or restrictions in the acquisition device can influence the measured shape of any finger(s) and the security level of FG systems may not be high [84]. To increase the performance of the FG, additional biometrics are generally employed in the proposed recognition schemes. For instance, single finger patterns of FV and FG were fused together in [87]. Another example, the palm print was combined with the FG in [28, 88, 89].

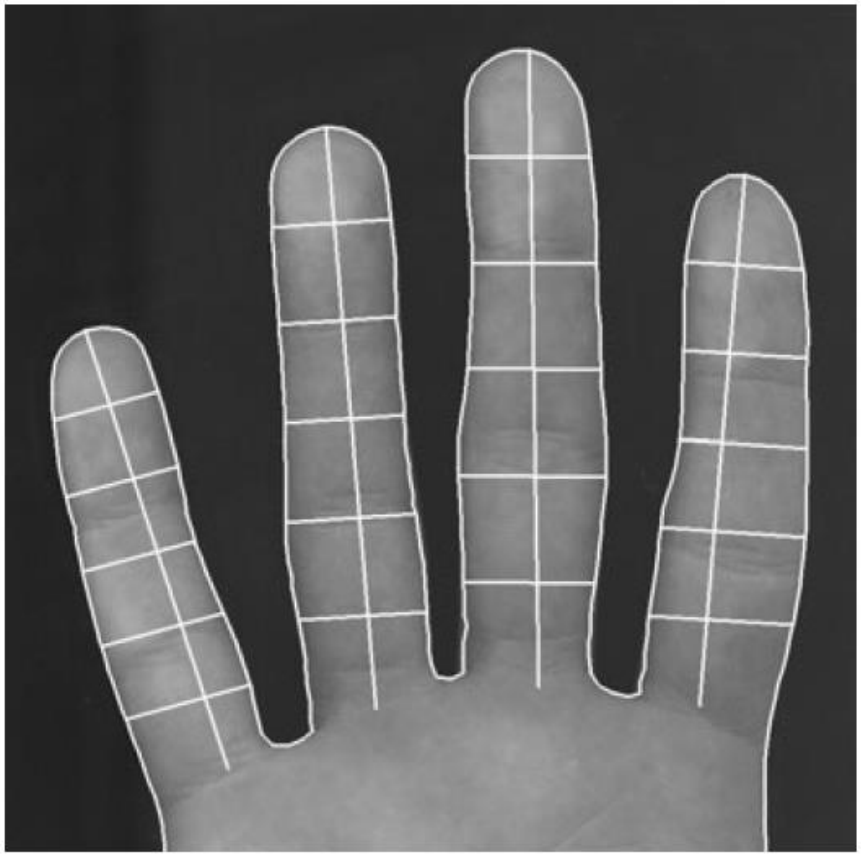


Figure 2.2: Common finger geometry features of the four fingers, where these are for each single finger: multiple finger widths in assigned locations and finger length [3]

2.2.3 Finger Veins

FV is a biometric pattern located inside the skin of a hand. This provides a high degree of confidentiality as it is so difficult to steal or use without a human's awareness. Furthermore, it can be only acquired when the individual is alive and this increases the spoofing difficulties of this biometric type [10]. Such a biometric requires a specific Near-InfraRed (NIR) environment. Fig. 2.3 describes a simple acquisition device of FV. It consists of NIR Light-Emitting Diodes (LEDs), a camera and NIR filter. The advantage of the white acryl in this figure is to equalize the lighting of the LEDs and to support the finger location [4]. Moreover, it has been cited that changing the position of the camera around a finger offers different views of veins and this can be beneficial to capture various FV patterns [90].

One of the main FV biometric problems is the difficulty of capturing distinct vein features. Therefore, invisible patterns or wrongly located fingers can increase the

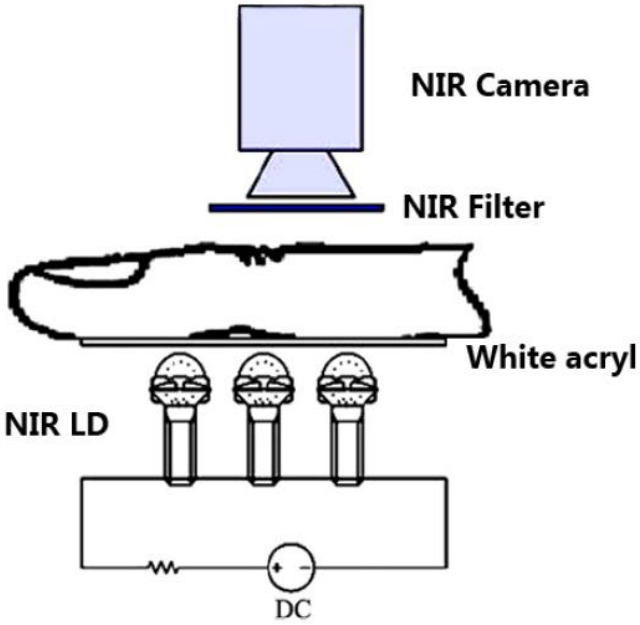


Figure 2.3: Example of an acquired FV environment [4]

false recognition performance [91].

Samples of FV images belong to the same person from the Hong Kong Polytechnic University Finger Image (PolyUFI) database (Version 1.0) [92] are given in Figs. 2.4, 2.5, 2.6 and 2.7.

It can be explained that available FV acquiring devices offer capturing one image for a single finger at a time such as [4, 10, 87, 93, 94], except in [90] where two camera devices have been used to simultaneously collect two finger images (one from each hand). The index finger, and sometimes the middle finger, is usually used in these devices. Furthermore, some of the captured FV images cannot cover the whole finger as in [87, 94]. So, not all of the possible FV patterns for all fingers are provided. Therefore, it could be argued that the obtained recognition

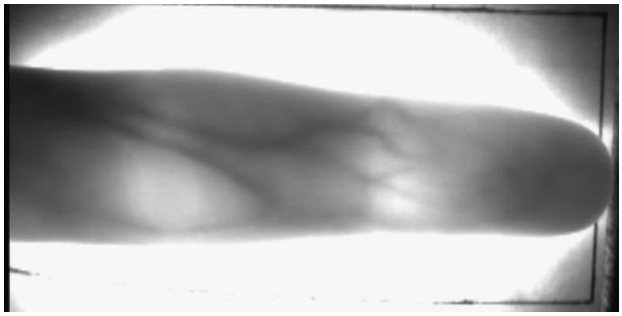


Figure 2.4: FV image of an index finger

Figure 2.5: FV image of an index finger shows parts of veins disappeared because of over illumination

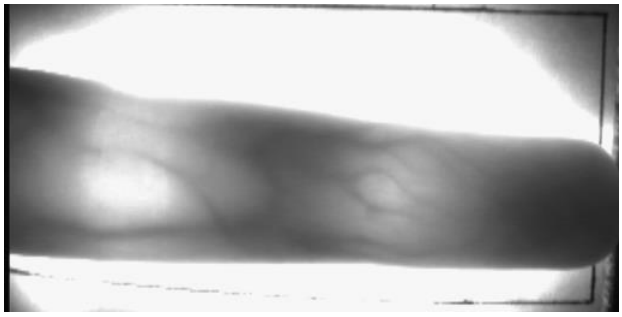


Figure 2.6: FV image of a middle finger

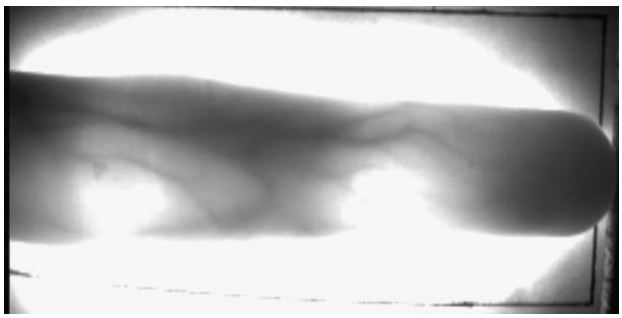


Figure 2.7: FV image of a middle finger shows parts of veins disappeared because of over illumination

performance can be enhanced if more patterns of veins were included. In addition, a question can be raised: what about amputating a certain finger?. In this case the biometric system could be useless for that individual.

2.2.4 Finger Outer Knuckle

FOKs are unique and reliable patterns. They exist in the dorsal finger surface positioned on the joint locations between the phalanxes. Principally, the FOK which is located between the distal and intermediate phalanxes contains the

'minor' features and the FOK which is positioned between the intermediate and proximal phalanxes includes the 'major' features [95]. The FOK pattern is believed to be distinctive and varies between the fingers rather than the individuals. One interesting observation which should be stressed is that the FOK offers different texture views according to various bending degrees. That is, bending the fingers around a handle knob of a door reveals valuable features [96]. Whereas, bending a finger around a peg to approximately 120° by using a special acquisition device offers clearly exploited textures [97, 98]. On the other hand, unbending fingers provides completely different FOK patterns [99].

Now, several recent publications, such as [97, 98, 100–102], have employed the Hong Kong Polytechnic University Finger-Knuckle-Print (PolyUFKP) database [103] to develop a new biometric identifier based on the FOKs. The acquiring device of this database, which has been designed to collect the FOK image, has a single peg with a specific angle to restrict the finger in suitable bending degrees (as mentioned, approximately 120°). Some Region of Interests (ROIs) images of bending FOK patterns are exhibited in Fig. 2.8.

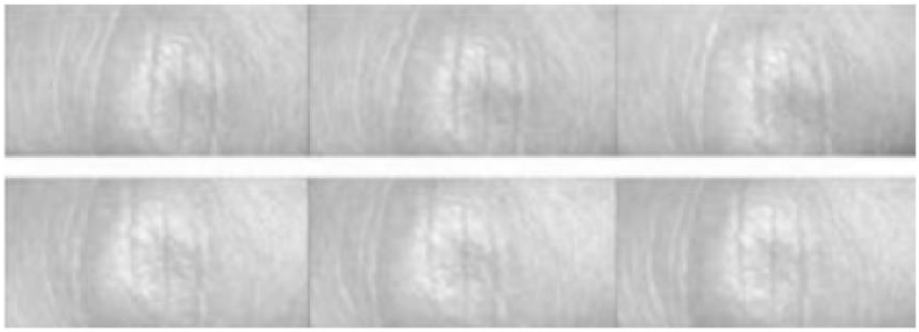


Figure 2.8: Some ROI images of the FOK 'major' pattern from the PolyUFKP database as shown in [5]

There are several difficulties associated with this database. First of all, if a certain acquiring finger is wrongly located in the capturing device, it could lead to an incorrect verification decision as explained in [97]. Secondly, the database has been collected to include only 'major' features for just two fingers (middle and index fingers), so, it includes limited features and overlooks many useful outer knuckles.

On the other hand, biometric systems have been designed in [99, 104, 105] to capture FOKs in an unbending situation by using a camera, which is located at

the top of an acquisition device. The 'major' features of three fingers (index, middle and ring) are collected in [104] and also the 'major' features of four fingers (index, middle, ring and little) are extracted in [99], whilst, all the 'minor' features were neglected. More concentration was considered for the 'major' features of just middle fingers in [105], as in the Indian Institute of Technology Delhi Finger Knuckle (IITDFK) database (Version 1.0) [6]. See Fig. 2.9.

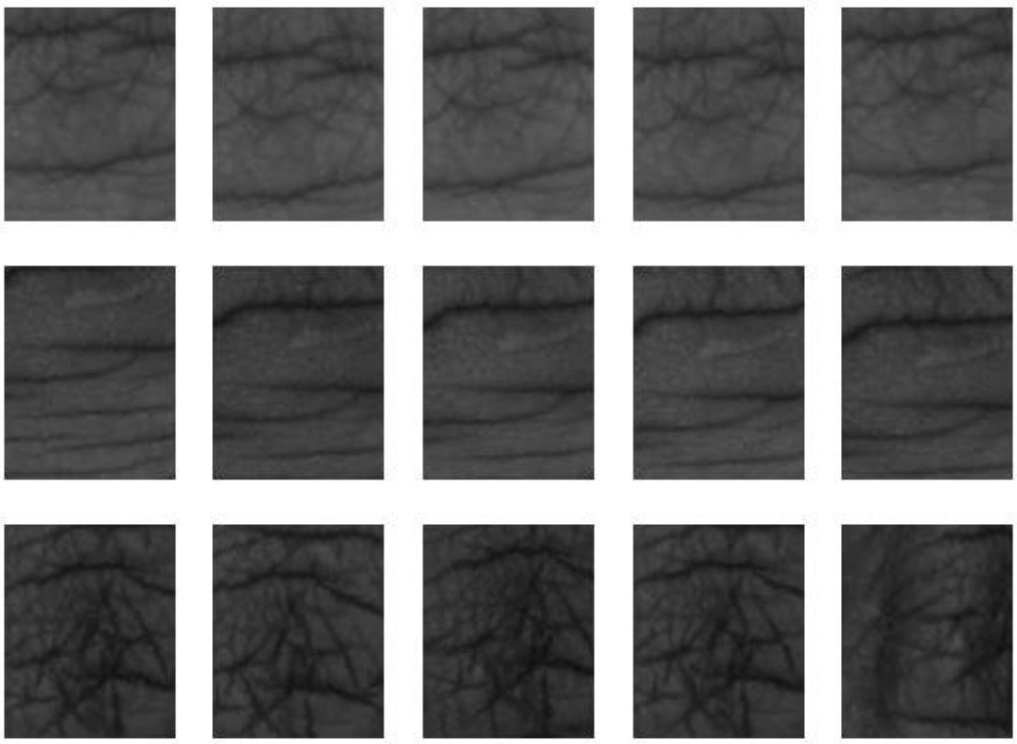


Figure 2.9: 'Major' features of three different individuals [6]; each row represents a 'Major' feature of a participant and each column represents offered samples from the different participants

Similarly, just the FOK of the middle finger was used, but employing the 'minor' and 'major' features in [95]. The reason beyond focusing on the middle finger as explained in [105] is that it provides satisfactory constancy during capturing the finger dorsal image and it also has a wide pattern area. Now, middle finger images have been provided as a database within the Hong Kong Polytechnic University Contactless Finger Knuckle Images (PolyUCFKI) database (Version 1.0) [106]. In this database, a long time interval of (4-7 years) between two sessions was recorded to collect the finer images. The reliability of the FOK patterns for the middle finger has been investigated and confirmed in [95]. Nevertheless, again it can be argued that using FOK pattern(s) of a single finger is(are) not enough because this

finger may be accidentally amputated.

The drawbacks of the unbending FOK patterns can be described as follows: firstly, their designed systems require a particular environment with fixed specification measurements [99, 104, 107]. To explain, acquiring the dorsal finger images requires a large box, a camera located at the top of the box, a hand position that is assigned at an appropriate distance at the base of this box, suitable lighting inside and an open slot to allow any size of a hand to go through it. Furthermore, the finger pose changing can directly influence the dorsal FOKs [108]. An additional problem which can be considered is that the FOKs do not have a normal protection like the inner FTs as they are located on the outer surface of a finger.

2.2.5 Finger Inner Knuckle

FIKs represent the flexion wrinkle patterns of the fingers that are clearly seen on the inner surface. Three knuckles can be recognized in each one of the main fingers (index, middle, ring and little), namely from the finger base: lower knuckle, middle knuckle and upper knuckle. So, the upper knuckle is the nearest knuckle to the nail, the lower knuckle is positioned on the base of the finger and obviously the middle knuckle is the FIK between the upper and lower knuckles. Fig. 2.10 illustrates the names and locations of the FIKs within a finger.

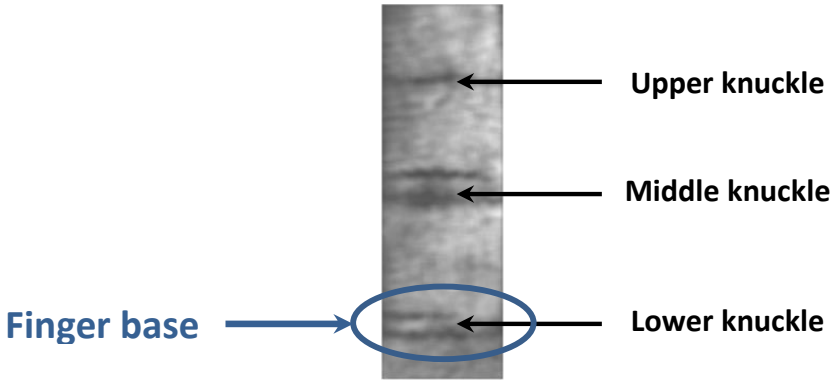


Figure 2.10: The locations of the three inner knuckles: upper, middle and lower

It is noteworthy stating that the prior work of this subject has excluded the inner knuckles of the thumb. One can argue that a thumb has rich FIKs especially the lower one. Moreover, to the best of the obtained knowledge not all of the three FIKs (the upper, middle and lower knuckles) were employed in the previous

studies except [109]. For example, just the middle knuckle of middle fingers has been utilized in [8]. The advantages of using the middle knuckle were explained in the same paper as it has a richer pattern than the upper knuckle and in comparison to the lower knuckle it is further stable. Also, the middle knuckles have been considered for three fingers (index, middle and ring) in [60] and for the four fingers in [110]. Middle and upper knuckles of the four fingers have been employed in [111], where a mobile camera was used to acquire hand images under different conditions such as different backgrounds and lightings. On the other hand, lower knuckles of the four fingers have been exploited in an innovative palm segmentation in [112]. Then, this work has been extended to collect the lower knuckles in order to be used as a biometric in [113] and they have been fused together with the palm print in a multi-modal biometric identification approach [114].

In the case of obtaining the FIK images, all the mentioned works have established their own database, so, they are not available. In contrary, a database was created from contactless fingers restricted by a backplate and a peg. The details of establishing this database have been described in [7], where parts of FIKs (the middle knuckles of ring and middle fingers) have been collected. From the same database, only the middle knuckles of the middle fingers were used in [115] and just the middle knuckles of the ring fingers were employed in [116]. So, a single FIK was considered from one or two fingers by [7, 115, 116]. Various patterns of inner middle knuckles are given in Fig. 2.11.

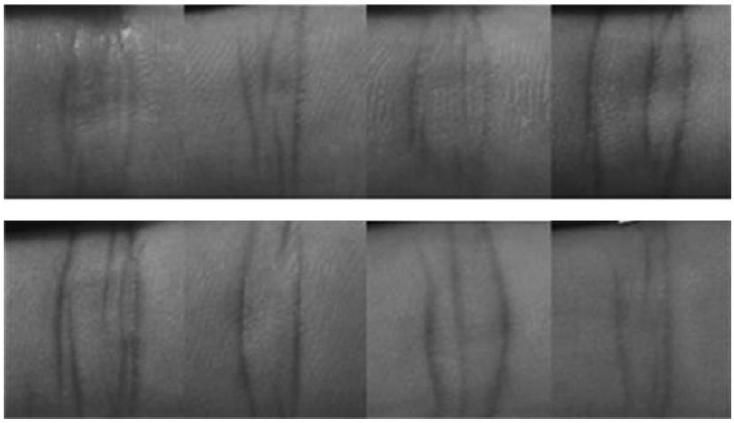


Figure 2.11: Samples of middle FIKs as demonstrated in [7] show various patterns of inner middle knuckles

The FIK has significant facilities over other biometrics, these are: it is easy to extract as it essentially consists of vertical lines; it has simple features and it can be recognized from low contrast and low resolution pictures [8]. In addition, comparing to the FOK it is less influenced by the finger pose and more stable [109]. On the other hand, the FIK studies have problems with: focusing on a limited number of knuckles; completely ignoring all thumb knuckles and neglecting rich textures of phalanxes, which are close to the ROIs of the used FIK(s).

2.2.6 Finger Texture

The inner surface of the finger has various rich patterns. These are: ridges, visible lines and skin wrinkles. The ridges require higher resolution images than the lines and wrinkles. So, the visible patterns of the flexion lines and wrinkles have simple and effective features [8]. The main features of the inner finger surface are expressed in Fig. 2.12.

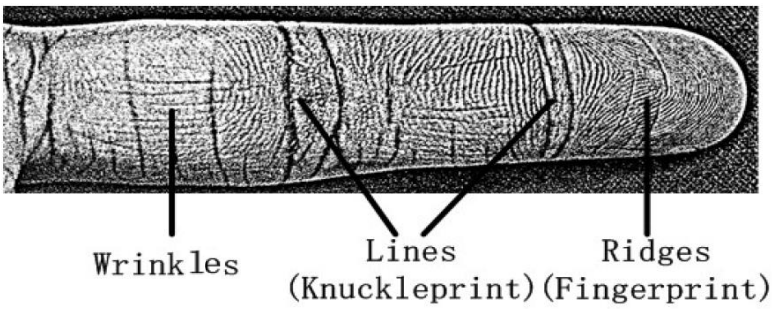


Figure 2.12: Various patterns that formed the inner surface of a finger: ridges, visible lines and skin wrinkles as given in [8]

Basically, the essential FT parts are phalanxes and knuckles. Three types of phalanxes can be recognised, these are a distal phalanx at the top of the finger holding the nail, an intermediate phalanx in the middle of the finger and a proximal phalanx near the base of the finger. The inner knuckles which create the clearly visible lines of the finger involve the upper knuckle between the distal and intermediate phalanxes, the middle knuckle between the intermediate and proximal phalanxes and the lower knuckle at the base of the finger. The FTs can offer a robust recognition performance as they have different patterns in each part. The principle parts of the FT of a single finger are demonstrated in Fig. 2.13.
All the FT patterns (excluding the ridges) can be obtained from low cost devices

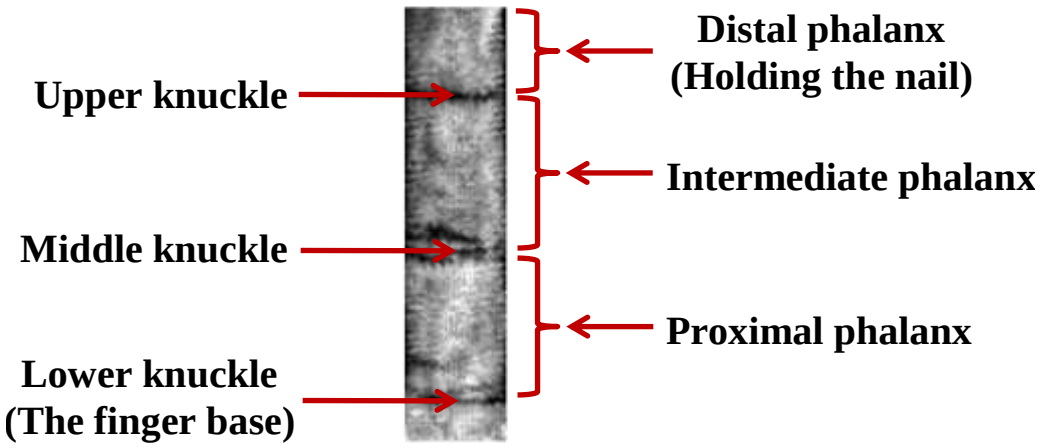


Figure 2.13: The fundamental parts of FTs of a single finger

and contactless hand images. Moreover, the ridge images for full FT regions are not available as a database, thus, they have been discarded.

There are many facilities that encourage the use of the FT as a powerful biometric. That is, they have rich information; unique between individuals or even between identical twins; easy to access; can be acquired without contact; resistant to tiredness and emotional feelings; their features are reliable and stable [64]; are clearly visible; so, they require an inexpensive low resolution camera (or scanner) to capture their images; they have normal protections as they are located in the inner surface of the fist [44] and it has been highlighted that FTs will not change over time, even for individuals who play racket-based sport such as tennis, despite such people physically using their inner fist muscles to grasp the racket [117].

FT based biometrics have attracted significant attention such as in [117, 118]. The FT characteristics are distributed among the five fingers. So, the five fingers of a hand can contribute together to give precise recognition decision. If an accident happens to any finger, there will still be four fingers present and they can be gathered to give good recognition performance. Therefore, a multi-object biometric system based on the FTs of the finger objects can be created.

In contrast, this subject requires more investigations as it was not always fully employed and it is usually combined with other biometric such as the palm print. More explanations will be given in the next section, which includes the literature review of this phenomenon.

There are some drawbacks regarding employing the FT as a biometric

characteristic. These are as follows: skin disease, dirty inner finger surface and amputating a part or full finger. Nevertheless, the amputating issue is considered in this thesis.

2.3 Related Finger Texture Literature Review

2.3.1 Segmenting Fingers and Extracting their Textures

The idea of employing the FTs probably started by Ribaric and Fratric [117], where this study introduced a multi-modal biometric system by using eigenfinger and eigenpalm characteristics. In this publication a scanner device was used to acquire high resolution hand images. So, the hands of the participants were located in a specific position with small limitations for finger translations. In this study a fixed ratio size of the FT is considered in each finger. This means that not all the FT regions were covered because the fingers have different sizes.

Ferrer *et al.* [29] proposed a low cost fusion system between the palm, hand geometry and FTs. The hand images were acquired from a commercial scanner with a resolution of 150 dpi. A simple contour was utilized after the image binarization by using the Otsu threshold [119]. Then, cartesian coordinates of the contour were converted to polar coordinates in order to calculate the locations of the tip and valley points of the fingers from the middle point of the hand base, where a hand was located in a specific location. The ROIs were the full finger images except parts of the lower knuckle patterns, which were ignored. The thumb was not considered too. Noticeably, not all patterns of the FTs were applied, high resolution images were employed and the most important thing is that a fixed location was used to acquire the hand images which did not allow free translations or orientation movements to be examined.

Ying *et al.* [120] adopted a hand parts segmentation method called Delaunay triangulation. The basic idea of this method is to simulate the hand image in a group of triangles to avoid the circular shapes. This study investigated the effects of different hand poses. It could segment the palm print and the five fingers of normally stretched hand parts (completed hand area should be included in the image) and failed with the others. This is due to how successfully the hand contour is established. This work discarded the lower knuckles, where the ROI of a FT was

determined by specifying 80% from a finger area. Besides, each ROI had different resizing area according to the mean sizing for the certain finger and this is not applicable because it had been calculated according to a specific group of images.

Pavesic *et al.* [118] established a study to fuse the fingerprints with the FTs for each finger. The authors also used a scanner acquisition device to collect part of hand images. So, they used high resolution part hand images located in a limited space, which allowed just small finger movements. The thumb was neglected with parts of upper and lower knuckles of the four fingers (index, middle, ring and little). Furthermore, the authors assigned each finger a fixed ratio size of the FT region, which may cause parts of the finger width to be cancelled.

Michael *et al.* [44] proposed a finger segmentation method based on the projectile approach, where a system was designed to track the five fingers from the hand video stream. Similarly, Michael *et al.* [45] presented the same segmentation method, however, it was implemented for only four fingers (index, middle, ring and little). The main problem of the projectile approach is that it could not detect very small distortions, rotations and translations as reported in [121]. Furthermore, the authors cited that larger rotation movements cannot be discovered too and they considered these would be unnatural behaviour for their acquisition device [44].

Kanhangad *et al.* [58] suggested a method to segment the four fingers based on the following steps: applying the Otsu threshold [119] for the binarization, employing opening morphological operations, implementing a simple contour, specifying tips and valley points for the four fingers by using the local minima and local maxima, determining four points in both sides of each finger to specify the finger orientations and then extracting the FT regions for only four fingers by using a largest adaptive inner rectangle technique. This process is useful for some transformations in the image plane as cited by the authors. Thus, it is not efficient for various hand rotations. The lower knuckles were discarded and they involve important features as confirmed in [25].

Bhaskar and Veluchamy [64] suggested a multi-modal biometric verification system based on feature fusion between the FTs and palm prints. This study used the Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66], but did not mention segmentation either for the palms or for the fingers. Moreover, it did not describe the partitioning of both training and testing sets.

From the literature, a simple contour operation was usually applied. Then, the

main finger points (tips and valleys) were determined based on the local minima and local maxima process. Obviously, these operations are not appropriate to address segmenting the fingers from the noise which may occur and from the hand images under various positions. Also, so far it can be noticed from the literature that there are serious issues about extracting just parts of the FTs. These weaknesses are required to be addressed and they are considered in Chapter 4 of this thesis.

It is worth highlighting that the first author's publication regarding the finger segmentation can be found in [25], where a method was proposed to extract the full regions of the FTs from the four fingers based on the traditional contour. This publication confirmed that increasing the collected FT patterns would increase the performance of the biometric recognition by using the Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67].

2.3.2 Prior Work on Feature Extraction

In the case of feature extraction, Local Binary Pattern (LBP) types have been found as efficient methods to be utilized. Essentially, they have been widely used in the face texture recognition such as [122]. However, they are also convenient to be adopted in the FT approach. This section will highlight the literature of the LBP method and the related prior work that has been stressed in comparisons.

The LBP method was firstly introduced by Ojala *et al.* [123] as a promising method for texture analysis. The main idea of the LBP operator is that each 3×3 neighbourhood pixels of an image, also known as an operator, are thresholded by the pixel in the centre. This will produce a binary or logical expression in each neighbourhood pixel. Then, this binary code is converted to a decimal value, which will represent the new centre value and it will be within the range of [0, 255] [123]. Consequently, the LBP was developed by Ojala *et al.* [124], where it was enhanced to consider multiple grayscale resolutions. In addition, rotation invariant micro textures were analysed and uniform LBP patterns were determined. However, the important characteristic which has been utilized in this thesis is the Multi-Scale Local Binary Pattern (MSLBP) or as called in the paper the multiresolution grayscale. After that, many approaches have been suggested to develop the

performance of the LBP method.

Jin *et al.* [11] proposed the Improved Local Binary Pattern (ILBP) operator for face detection. The key idea of this operator is including the centre pixel in the LBP function and assigning a high significant weight to its value. In this case the information of the centre pixels were exploited as they could be very important.

Tao and Veldhuis [125] described a Simplified Local Binary Pattern (SLBP) operator as an applicable method to solve the illumination variation problems. This method was applied and evaluated in a face verification system. Its operation was suggested to remove any multiplication weight from the LBP function. Therefore, it was reported to be very quick, easy to implement and produces directionless textures.

Complicated LBP methods have been proposed by Wolf *et al.* [12] called Three-Patch Local Binary Pattern (TPLBP) and Four-Patch Local Binary Pattern (FPLBP) respectively. These two novel operators were used for the face matching between pairs of images, to classify them into belonging or not belonging to a certain person. The two methods have been proposed to apply the LBP process for a patch of pixels instead of a small matrix of pixels. The TPLBP indicates three patches of pixels and the FPLBP refers to four patches of pixels. The global textures will be maintained while the micro textures are mistreated and this seems to be a major problem. Furthermore, increasing the number of patches (from three to four) disregards more small features.

Petpon and Srisuk [126] derived a Local Line Binary Pattern (LLBP) from the original LBP by considering lines of pixels in horizontal and vertical directions only. Then, similar LBP operations are applied to each vector. To combine the obtained LBP codes from the horizontal and vertical pixels an amplitude equation was implemented. This structure was employed for a face recognition issue, where face characteristics were analysed as vertical textures, horizontal textures and then a combination between the vertical and horizontal textures.

Ahmed [127] has suggested a descriptor that is called Gradient Directional Pattern (GDP). He applied this descriptor for recognizing facial expressions. This study was aimed to overcome the sensitivity of the LBP towards illumination changes and noise in the face expression image. The operations of this descriptor start with using an average filter of size 5×5 pixels, separately detecting horizontal and vertical edges by convolving the face image with the Sobel operators, calculating

the angles of the gradient directions in degrees, implementing a GDP operator after setting up a threshold to suppress undesired information, partitioning the face image into determined block sizes, computing the histogram of each partition, concatenating the resulting vectors to one feature vector, employing a support vector machine to classify the expression and computing the recognition performance by utilizing the cross-validation technique.

The Improved Local Binary Pattern Neighbours (ILBPN) structure was proposed by Liu *et al.* [116] to analyse FIKs. This scheme consists of the following steps: Gabor filtering, mean filtering and feature analysing by applying a special LBP operator, which consists of a horizontal vector only. Then, uniform values from the resulting image have been established for each pixel and binary images have been produced for each uniform value. Limited features have however been utilized in this publication, because just middle knuckles of ring fingers were utilized.

Tong *et al.* [13] developed an operator termed Local Gradient Coding (LGC) and derived an additional operator named Local Gradient Coding-Horizontal Diagonal (LGC-HD). Both operators were examined in the case of facial expression recognition. The LGC operator idea is to implement a gradient difference around the centre pixel. So, three main directional differences are expected: gradients around the vertical, horizontal and diagonal. Then, these gradients are used to compute the LGC code according to calculating the original LBP code. The authors investigated the effects of the vertical gradients and they noticed that their influences were not significant to their employed database. Furthermore, considering their analyses will be time consuming. Therefore, the LGC-HD has recommended to reduce the analysing time and enhance the expression recognition performance by removing the redundant information. It is true that the LGC-HD reported better performance than the LGC in [13] according to the presented application, but this might not work well with other applications. Both operators have been used for comparison purposes in this thesis.

The GDP method is inspired in terms of employing the gradient directions of the edges. On the other hand, some essential drawbacks can be investigated. For example, the threshold that is used in the GDP operator may remove some important information and it is empirically tuned or discovered for each database, so, it is not adapted to an undesired input image. Furthermore, employing the histogram to collect the feature vector has problems of establishing a very long

vector and as highlighted in [116] effective spatial data can be wasted by applying this method. The LLBP method seems to be suitable for analysing the FT features as they generally compromise vertical and horizontal patterns. Nevertheless, the combination between the codes of both horizontal and vertical features can be enhanced. These issues will be illustrated in Chapter 5, where two feature extraction methods are proposed.

2.3.3 Literature Review of Multi-Object Fusion

Although extensive work has been carried out for single-modal biometric systems, no intensive study exists to concentrate on the inner finger surface textures as a multi-object biometric prototype. Generally, several studies have documented FTs as a part of multi-modal biometric recognition.

Ribaric and Fratric [117] presented a first investigation about combining FTs with palm prints in the case of human identification. In this work, the contribution rate of each finger was calculated and a score fusion between the five fingers and palm was implemented. This fusion was based on the weighted summation rule.

After that, a combination of palm, FTs, and hand geometry was used to produce a low cost multi-modal biometric recognition system by Ferrer *et al.* [29]. In this study, different types of fusion were evaluated: decision fusion with the voting rule, score-level fusion by the weighted summation and feature fusion based on the two-dimensional convolution. It was recorded that decision fusion obtained the most satisfactory results.

Ying *et al.* [120] evaluated two fusion methods between FTs of five fingers and the palm print. The first method was based on the feature level fusion termed holistic, where all the regions of the FTs and palm print were exploited in the same feature extraction method. The second combination method was suggested to use a weighted summation rule as a type of score fusion, where hand parts (palm print and FT of each single finger) were processed separately then the score fusion was obtained after applying the Hamming distance matcher. Finally, the score fusion was found to perform better results than the holistic method.

Nanni and Lumini [128] suggested a fusion of multiple matcher scores by using the summation rule. Two different processes were applied to the same FT, then, fusion between the matching scores in the final processing stage is applied. Firstly, the

proposed approach was implemented for middle fingers by considering that the full segmented finger image is the FT region. Subsequently, ring fingers were augmented to improve the recognition performance.

Pavesic *et al.* [118] produced a comparison study for fingerprints and the FT surface based on principal component analysis, the most discriminant features and regularized-direct linear discriminant analysis. The authors chose eight regions from four fingers (index, middle, ring and little) to be used in both verification and identification. The two main regions were the digitprints, which representing the FT of each finger, and the fingerprint, as the upper regions of each finger. Then, the score fusion of all the employed finger parts was considered according to the weighted summation rule. In terms of feature extraction, the best results were reported for the regularized-direct linear discriminant analysis method.

Michael *et al.* [44] illustrated a robust recognition system by combining the FTs of the five fingers and the palm print in the case of verification. No standard resizing was used for the FT regions. An automatic tracking algorithm was implemented to acquire the hand images from a device with a real-time video camera. In this work the FTs were combined with the palm print to achieve a robust verification. Again, a score fusion was utilized, but by applying the Support Vector Machine (SVM) technique with the Radial Basis Function (RBF) to implement the combination between the FTs and the palm print. Before that, a matching was applied by using the Hamming distance for the palm print and Euclidean distance for the FTs. Likewise, Michael *et al.* [45] applied the same fusion method, but excluding thumbs.

Kanhangad *et al.* [58] employed the FTs as a part of a fusion study between palm print, hand geometry and finger surfaces from 2D and 3D hand images to enhance the contactless hand verification. The weighted summation rule was also utilized in this work in terms of score fusion.

Zhang *et al.* [129] proposed a score fusion method by using the summation rule. The fusion was performed between two types of biometrics, the palm print and FT of only the middle finger. The image of the segmented middle finger was treated as the FT region. A two dimensional wavelet technique was employed to collect the features of both biometrics. The wavelet coefficients of approximation, horizontal details and vertical details were collected separately for each biometric. An average filter was applied just to the palm print coefficients of the horizontal and vertical

details. Consequently, feature level fusions were implemented on all coefficients of each biometric. Then, the score fusion was executed after the matching operations. Kumar and Zhou [10] examined vein and texture images for just two fingers (index and/or middle) in two experiments, where different numbers of subjects were employed for the personal identification. The major problem of this work was that the database was acquired for a small region of the fingers. In addition, just two fingers were employed (the index and/or the middle finger). This could be the reason why the authors used two innovative types of score fusion termed holistic and non-linear, both of which are mainly based on the vein features.

From the previous literature, it can be observed that the FTs have been used as a part of a multi-modal biometric scheme. In addition, limited FT features have been utilized as illustrated in Section 2.3.1. It can be argued that reliable biometric recognition approaches can be fully established by employing all the FT patterns of the five fingers and exploiting them in a fused multi-object prototype. This will reduce the cost of providing an additional acquiring device and establishing an extra applicable algorithm for including another biometric trait to be combined with the FTs. Fusion between multi-objects of FTs will be described in Chapter 6.

2.3.4 Related Work on Performance Measurements

The main classifier in this thesis is the Probabilistic Neural Network (PNN). It has been noted that there is a serious problem regarding establishing the Receiver Operating Characteristic (ROC) graph from this multi-classifier network alone. This is because the score values are not available in this network and these values are important to construct the parameters of the ROC curve, as will be explained in Appendix A. There is no prior work which could clearly consider this matter as shown next.

To start with, a new strategy to produce an ROC for an Artificial Neural Network (ANN) is proposed by Woods and Bowyer [130]. The essential idea of this work is to utilize the bias in the hidden layer to give positive and negative offsets and calculate the ANN outputs as scores. The problem of this study is that their contribution is just for a Multi-Layer Perceptron (MLP) neural network, which was designed for only two classes. The topology of this network should have one

input bias for only the first hidden layer, no bias connections for the other layers and a single output neuron.

Orr [131] evaluated the possibility for death following cardiac operation by employing a PNN. The author took advantage from the Probability Density Function (PDF) inside the neural network and predicted the postoperative death. A commercial software called "NeuroShell 2" was applied to construct the PNN and collect the ROC graph. However, the depicted ROC curve was not smooth and it was designed for a PNN with only two classes.

In Ooi *et al.* [132] a signature verification model was created by using the PNN. An interesting method named "strengthening" was used to strengthen the effects of signature templates. The key idea of this method is slightly varying the signature feature values according to a Gaussian distribution in order to increase the probability of the original data. This was the reason of how the ROC graph could be established, where the PNN was executed for many times with various information after the strengthening process. So after the re-executions, the output values were developed or, in other words, the score values were constructed.

Sharma *et al.* [133] proposed a modified PNN for the iris classification purpose. The resulting network was named Parzen PNN (PPNN), where the Parzen approach [134] was combined with the PNN. The heart of a PPNN is about partitioning the training data by using the Parzen window, then, applying these subsets to the PNN and considering their PDFs for each class. The network has the same PNN structure. Nevertheless, the output layer was modified to classify the information according to maximum function instead of employing the 'winner-takes-all' rule. It appears that the output values were utilized as score values to produce ROC and Cumulative Match Curve (CMC) graphs. The major problem of this method is increasing the network processing units. Therefore, an additional operation called kernel discriminant analysis was implemented in order to decrease the overall size of the input features.

Similarly, Joshi *et al.* [135] employed the PPNN for periocular identifications/verifications. The authors also used an additional feature reduction process after applying Gabor filter called the direct linear discriminant analysis. Again this was due to the long sequence of data resulting from the Parzen window. The outputs of the PPNN were utilized to establish ROC and CMC graphs. According to the authors the final results were collected from these graphs. It can

be observed that there are big differences between the recorded results of the identifications and verifications. Generally, the reported accuracies of the identifications were significantly worse than the recorded accuracies of the verifications. This is due to the output (or score) values of the PPNN, where disparities among them can be found. So, the influence of these disparities or variances clearly appeared in the identification comparisons between the values.

Almaadeed *et al.* [136] suggested an approach to identify speakers. Briefly, the approach consists of two parts: wavelet analysis for feature extraction and ANNs for classifications. The authors collected the ROC parameters by applying a combination of multiple neural networks. Those were the General Regressive Neural Network (GRNN), Radial Basis Function Neual Network (RBFNN) and PNN. A voting process was applied to the decision of the three neural networks and there values were utilized as scores. Obviously, this suggestion increases the complexity of the system.

It is worth mentioning that many other publications employed the PNN in the case of biometric recognition without establishing the ROC curve such as [40, 137–147].

It is clear that there is no specific method to construct the ROC graph from the multi-classifier PNN unless employing a combination process technique, which will cause more complexity and be time consuming. So, a novel method is presented to address producing the ROC curve from a PNN. In addition, generating the ROC graph has been developed in this thesis to also include a suggested innovation classifier named the Finger Contribution Fusion Neural Network (FCFNN). Specifically, addressing producing the ROC graphs will be clarified in Appendix A.

2.4 Summary

As rich textures exist in each finger, various features can be observed. The first very old and well know biometric is the fingerprint. While, other finger characteristics were recently investigated such as FG, FV, FOK, FIK and FT. Each one of these biometrics has valuable features. On the other hand, each characteristic has benefits and drawbacks. So, a general overview of the common finger characteristic patterns was provided.

Moreover, relevant literature review related to the FT as a type of biometric was stated. It can be noticed that the FT has not been intensively and thoroughly

studied before. As with each step of the personal FT recognition there are noticeable problems, the related prior work of each step was described. Firstly, for the finger segmentation and ROI extraction, the majority of the previous studies considered limited areas of the FT regions. In addition, many publications completely ignored the textures of thumbs. Secondly, FT patterns lacked a beneficial feature extraction model that can efficiently collect its horizontal and vertical features. Thirdly, usually the FTs were combined with supported biometric characteristic(s) and construct biometric structures of multi-modal. Whereas, in this thesis it has been demonstrated that this trait can be used alone as multi-objects by considering the FT of each finger as a single object. Finally, for the performance measurement, it can be investigated from previous studies which exploited the PNN classifier alone that they could not construct the ROC graph. This issue is considered and solved in this thesis.

The next chapter will explain the available databases of the FTs. Due to the fact that there are limited numbers of provided FT databases, full hand images in the palm print databases have been used.

Chapter 3

Database Overview

3.1 Introduction

This chapter concentrates on introducing the databases of inner finger surfaces from which the Finger Textures (FTs) are extracted. Although personal recognition based on FTs has had growing importance there are no specific databases for the full regions of FTs. Therefore, hand images which are acquired generally for their palm prints have been employed since FTs can be extracted from the fingers of these images. Moreover, the performance measures will be clarified in this chapter. In this study, three databases have been employed:

- The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67].
- Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66].
- Spectral 460nm (S460) from the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9].

Each of the used databases has specific characteristics and resolutions. Furthermore, the environment of acquiring the hand images in each database has been exclusively designed as will be described.
The remainder of this chapter is composed of: Section 2 describes the databases of the FTs and their challenges. Section 3 explains the basis of calculating the performance measures. Section 4 summarizes the chapter.

3.2 The Databases of the Finger Textures

3.2.1 The Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database (Version 1.0)

In this database, a commercially available 3D digitizer, Minolta VIVID 910 is used to capture 3D and corresponding 2D hand images. The database consists of 1,770 images from 177 different people, where each person has contributed 10 images. All the images are in colour and no fixed position or restrictions were imposed. The same indoor environment has been used with a black background to collect image data from the palm side. The images were captured in two sessions and five images were collected at each session. The time between the two sessions ranges between one week (only for 27 participants) to three months. The age range for the participants is between 18 to 50 years from students and staff with multiple ethnic backgrounds and both genders. Furthermore, the participants have been asked to take off rings or jewellery and they have been asked to make small movements or slightly change the hand position after each acquisition. All images are of type bitmap. Each hand image has a resolution of $640 \times 480 \times 3$ pixels and the participants were asked to locate their hands far from the scanner to approximately 0.7m [67]. Therefore, the hand images in this database can be considered as very low resolution. Examples of hand images belonging to the same subject are given in Figs. 3.1, 3.2, 3.3 and 3.4. Small hand movements can be observed between these figures.

A second version of this database named The Hong Kong Polytechnic University Contactfree 3D/2D Hand Images Database (Version 2.0), simply denoted PolyU3D2DV2, is available. This version considers the different poses of the hands. This database is not completely provided as only the data which has been collected from the first sessions can be downloaded, so it is not employed. On the other hand, another available database which have considered various poses of hand images will be explained in the next section .

3.2.2 IIT Delhi Touchless Palmprint Database (Version 1.0)

The IIT Delhi Database has basically been established to overcome the variation of the hand location drawbacks of other palmprint databases, where restricted

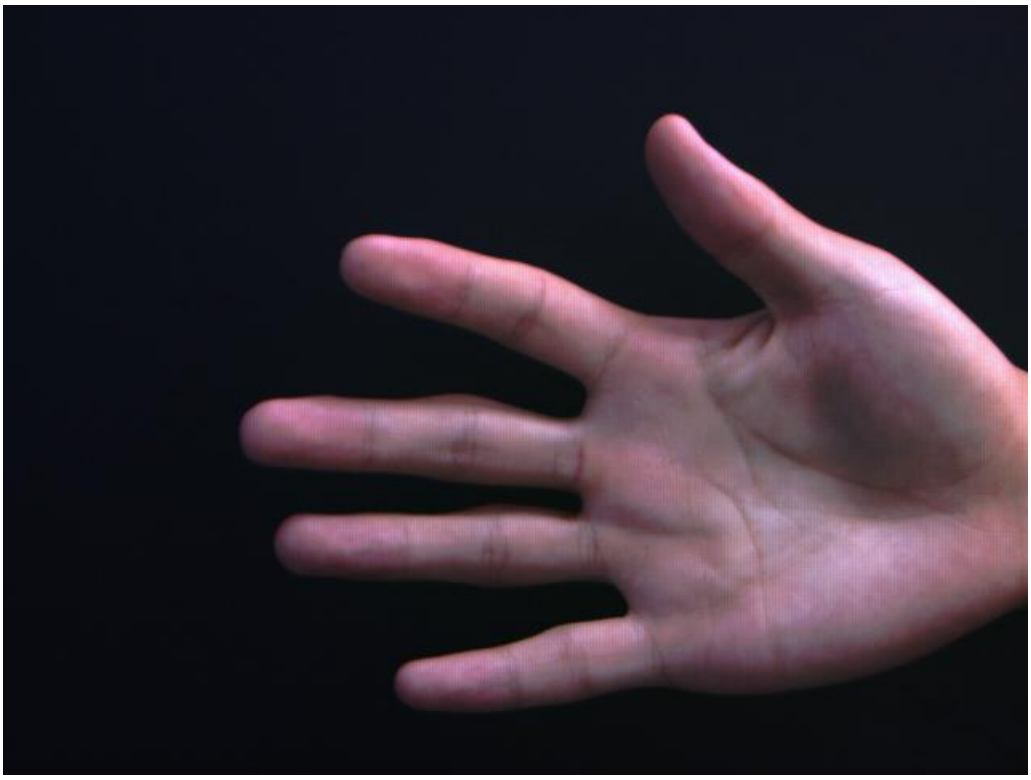


Figure 3.1: Example of a hand image from the PolyU3D2D database

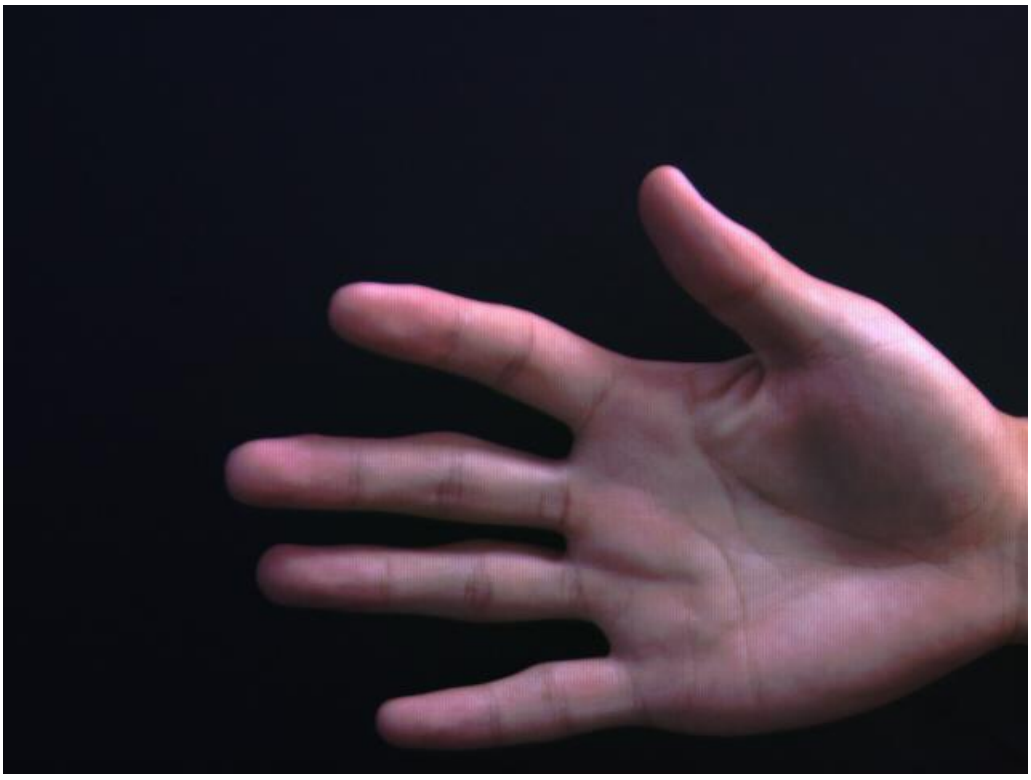


Figure 3.2: Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 0.2cm to the left

Figure 3.3: Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 1cm to the left

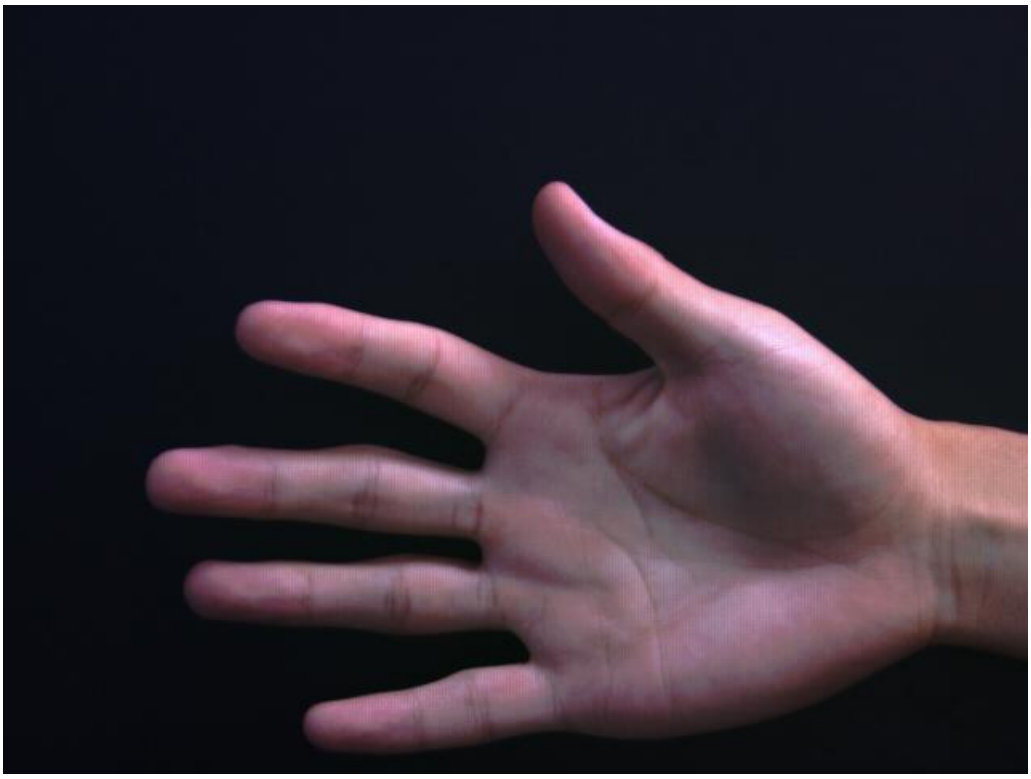


Figure 3.4: Example of a hand image from the PolyU3D2D database for the same subject as in Fig. 3.1 with a translation of approximately 1.75cm to the left

environments were designed to acquire palmprint hand images with fixed poses determined by pegs. It can be claimed that using the pegs to capture a hand image is very uncomfortable as the users are obliged to put their hand into specific location. The Biometric Research Laboratory in IIT Delhi/New Delhi/India has therefore designed a peg-free environment to capture groups of hand images and utilized different variations in order to expand the researching area in terms of trustworthy palmprint recognition. It was assembled over the period July/2006 until Jun/2007 from the IIT Delhi staff and students. This database has been provided as an open access database for the research area since October 2007. A simple design was facilitated to collect the data from the participants in a contact-free manner with high variations of movements. An indoor environment has been utilized to collect hand images from 235 participants between the ages of 12 and 57 years. Both genders were present in their hand images. An open camera was used to view the hand image before the capturing operation. The camera lens was surrounded by fluorescent lighting in a circular shape. A bitmap format is used to store the hand images [65]. The IIT Delhi database can be considered as high resolution coloured data, where each hand image has the size ($1200 \times 1600 \times 3$ pixels). The IIT Delhi Database consists of 1,603 right and left images. Just the right images have been focused upon in this thesis as the PolyU3D2D consists of just the right hand images. This will establish fair comparisons. In addition, asking the users to put both hands in the acquisition device is awkward and using two capturing environments will be expensive. However, not all measurements have enough samples to cover the neural network training and testing requirements. In particular, 888 right hand images have been used from 148 people, where each person has participated 6 images.

This database can be considered as the most challenging one in terms of the FT studying area, because it includes hand images with various postures according to [66, 148]. Figs. 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14 show hand image samples with abnormal situations such as bending finger(s) because of restricted space and distorted hand image.

Moreover, ring jewellery can be found on one or two fingers of hand images in this database which, as mentioned, has been generated basically for palmprints. Nevertheless, these exist in all samples of the corresponding participants. In other words, they appear as parts of the FT for the participant who wear the ring(s).

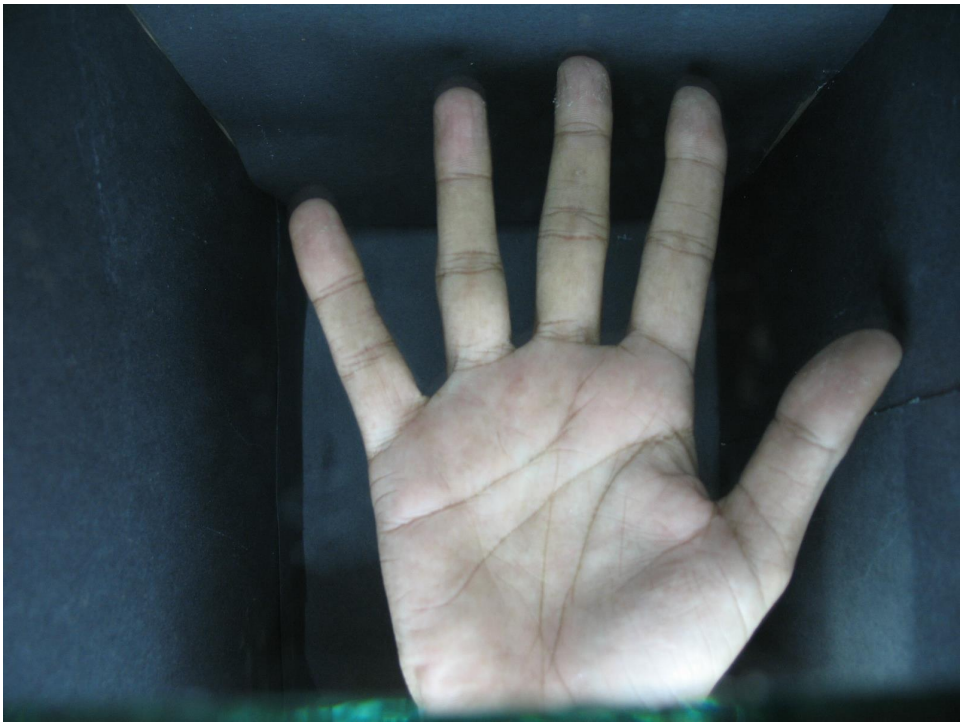


Figure 3.5: Example of a hand image from the IIT Delhi database

Figure 3.6: Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 10° to the left

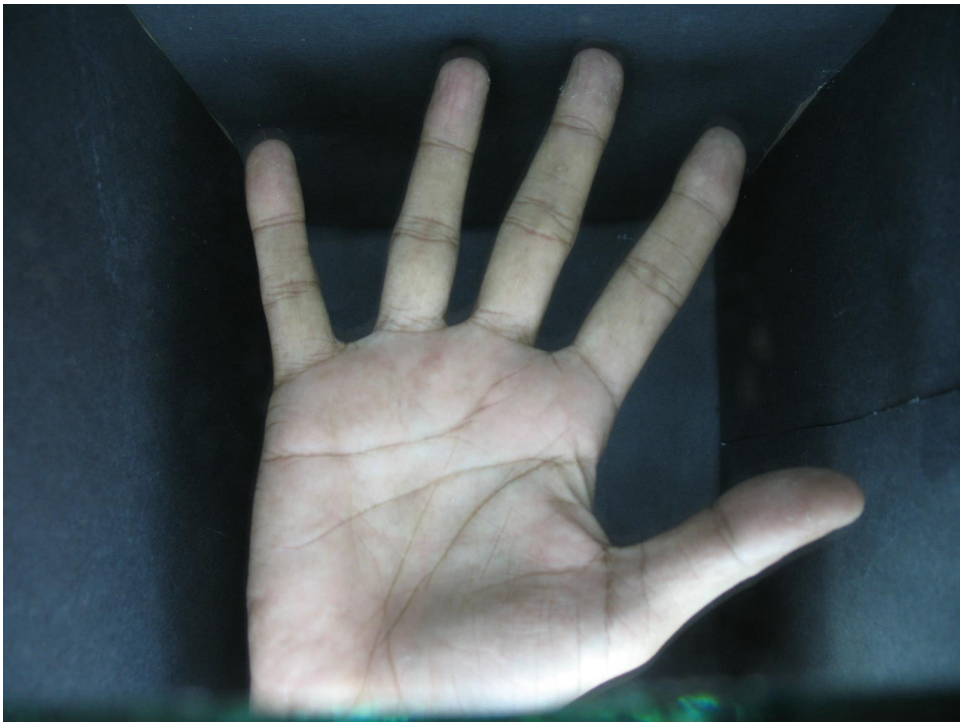


Figure 3.7: Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 20° to the right

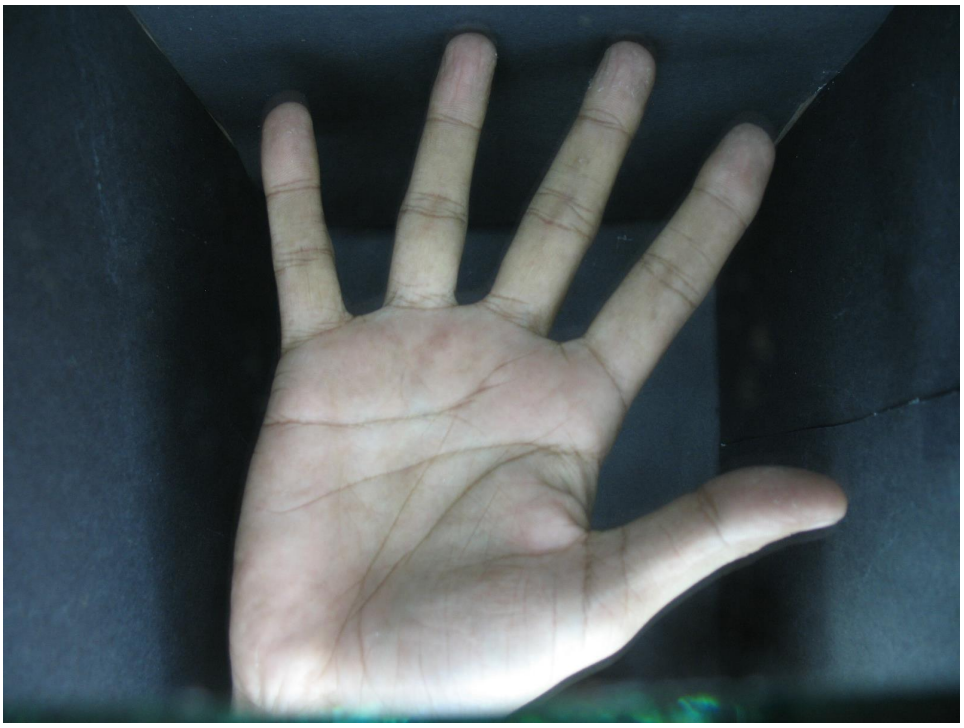


Figure 3.8: Example of a hand image from the IIT Delhi database for the same subject as in Fig. 3.5 with a rotation of approximately 30° to the right

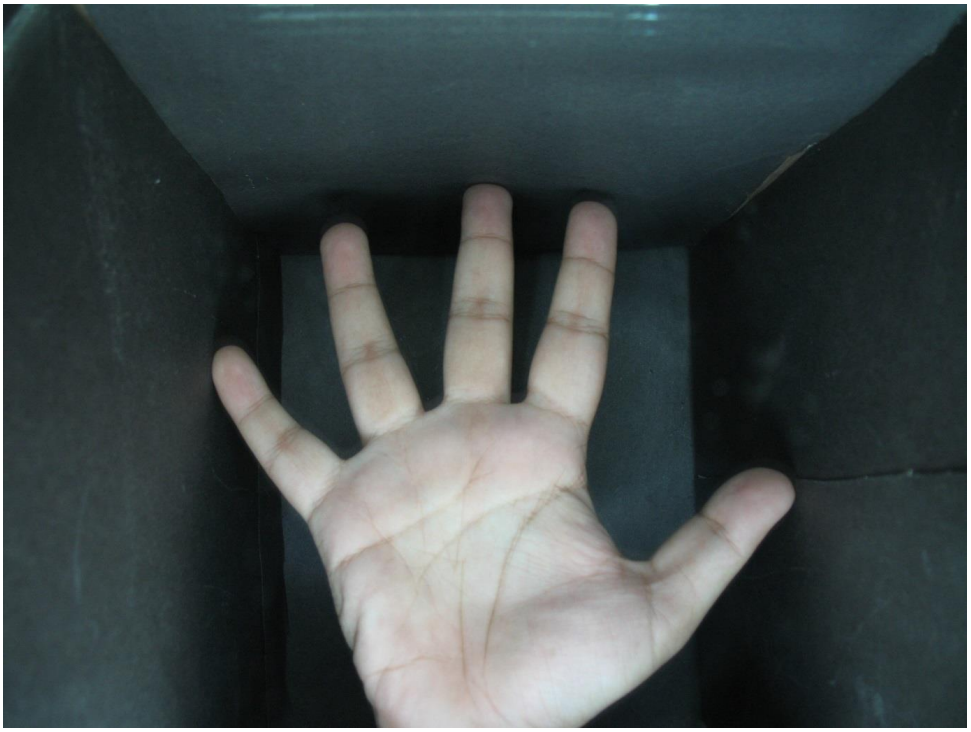


Figure 3.9: Sample of a hand image from the IIT Delhi database showing a middle finger bent to the back because of restricted space

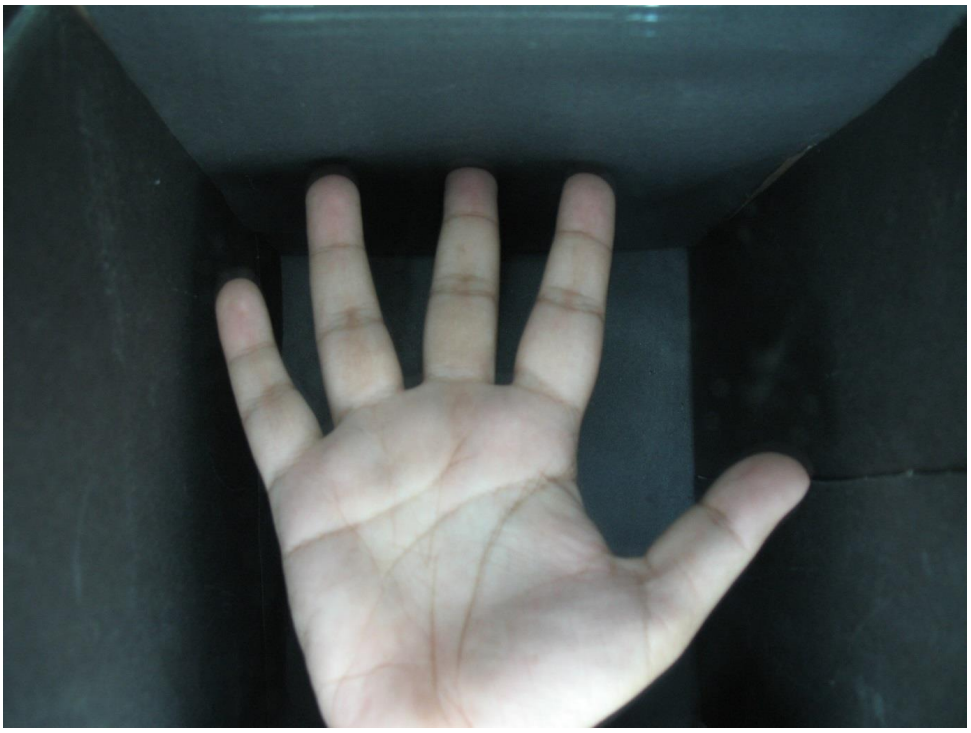


Figure 3.10: Sample of a hand image from the IIT Delhi database illustrating a middle finger more significantly bent to the back because of restricted space

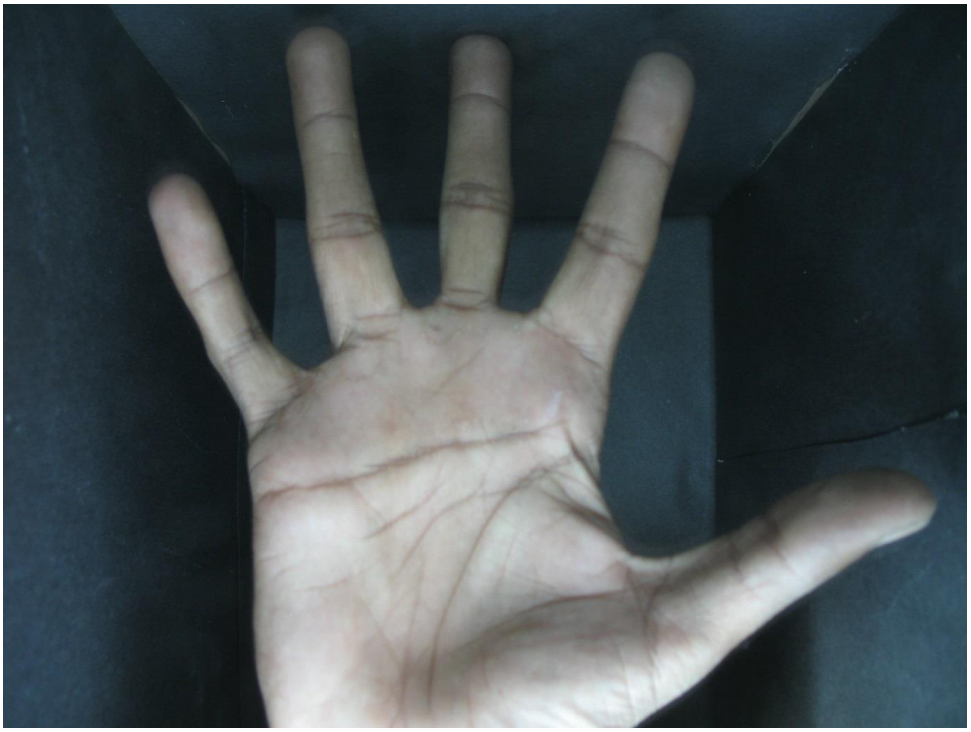


Figure 3.11: Sample of a larger hand image from the IIT Delhi database demonstrating a middle finger bent to the back because of restricted space

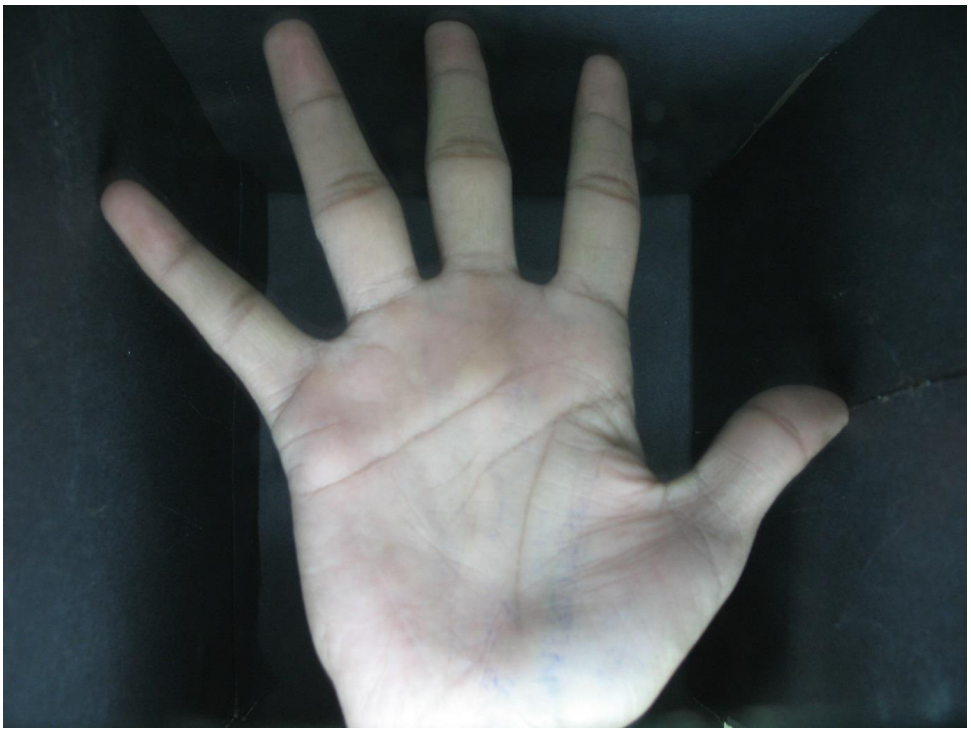


Figure 3.12: Sample of a large hand image from the IIT Delhi database with bent middle, index and ring fingers to the back

Figure 3.13: Sample of a hand image from the IIT Delhi database with a bent thumb and a rotated hand of approximately 25° to the right direction, where a small amount of texture from the thumb has been lost

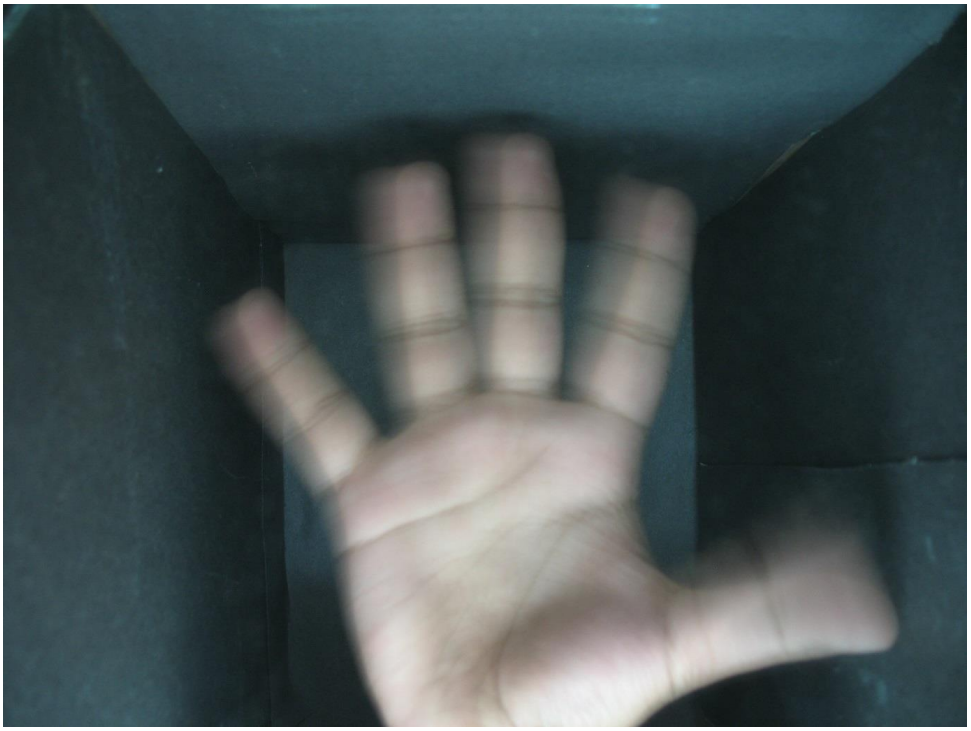


Figure 3.14: Sample of a hand image from the IIT Delhi database illustrating a distorted hand image because of instability during the capturing operation

Figure 3.15: Example of a hand image from the IIT Delhi database showing a gold ring appearing in the ring finger

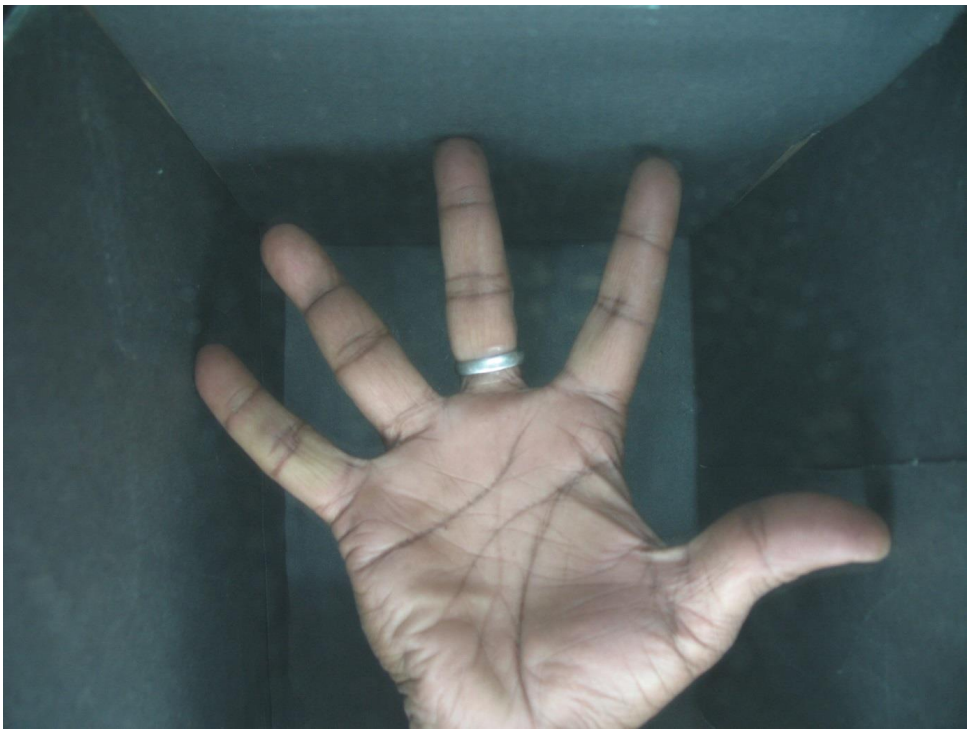


Figure 3.16: Example of a hand image from the IIT Delhi database demonstrating a silver ring in the middle finger

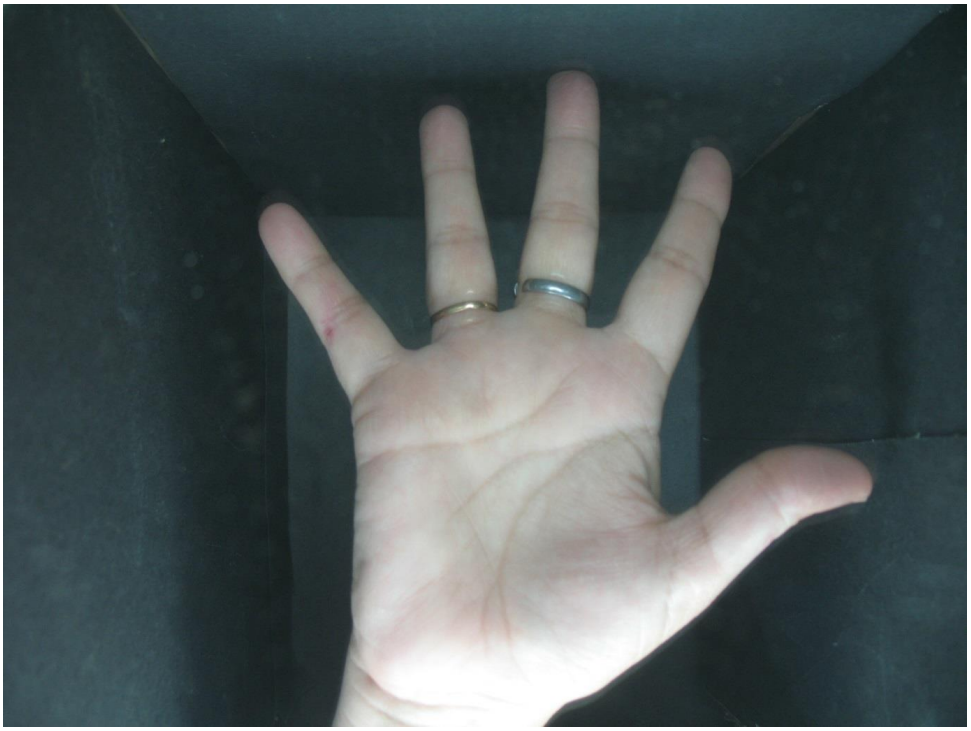


Figure 3.17: Example of a hand image from the IIT Delhi database with gold and silver rings on the ring and middle fingers

Figure 3.18: Example of a hand image from the IIT Delhi database showing two gold rings appearing on the ring and middle fingers

Examples of hand images where gold or silver rings exist can be found in Figs. 3.20, 3.21, 3.22 and 3.23. It is expected that their presence may affect the personal verification performance when using the FTs as a biometric, thereby representing a useful user case.

3.2.3 CASIA Multi-Spectral Palmprint Image Database (Version 1.0)

In this database, multi-spectral light sensors have been used to capture different features for the hand images. Principally, the skin of an inner hand surface shows various characteristics if different light spectra are applied. This is due to the penetration of the given spectrum. Therefore, identifiable patterns can be noticed under the skin of the inner hand surface after using a specific spectrum of lighting such as the veins. A multi-spectrum acquisition device was created to capture six types of patterns as hand images, see Fig. 3.19. These images have been made open access to expand the studies of biometrics. The participants hand is free to move in the acquisition device. It is peg-free as there are no limitations to the

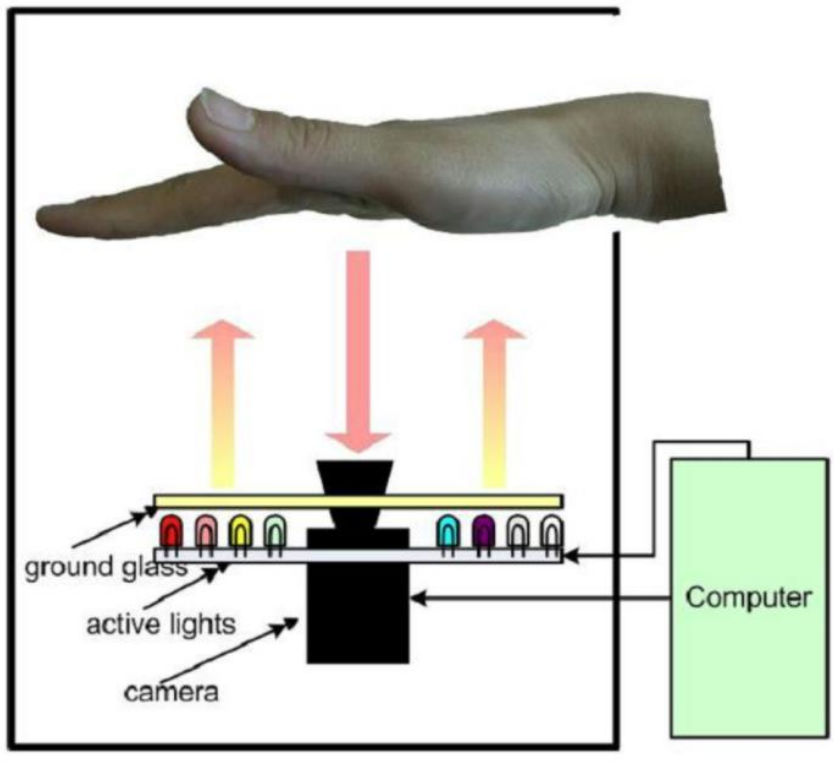


Figure 3.19: A demonstration of the multi-spectral hand image acquisition device for the CASIAMS database as given in [9]

location of the hand. However, the participants were expected to open their hands inside the acquisition box. A dark background (mainly black) was used. A Charge Coupled Device (CCD) camera was located at the bottom of the device and the lighting was equally distributed. A controller circuit was created to automatically activate the spectrum lighting. The specific ID of the individual was given as the name of each image file, where useful information is understandable from these names. A total of 100 users contributed with their right and left hand images. Six samples were captured in two sessions. That is, three samples in session one and after more than one month in session two an additional three samples were collected. Multi-spectral wavelengths, which were generated by the provided lighting, were used to capture six image patterns at one certain time. The utilized spectra had the wavelengths of 460nm, 630nm, 700nm, 850nm and 940nm. In addition to the white illumination. So, the total number of the provided CASIAMS images in this database was 7,200 right and left hand images. These were stored as Joint Photographic Experts Group (JPEG) images and they all are 8-bit grayscale. In the case of resolution, these touchless hand images can be considered as low resolution as each hand image has the size 576×768 pixels. The participants had been permitted to make determined hand movements in order to increase the reliability of the palmprint biometric studies, which utilized this database. Samples of the six multi-spectral hand images belonging to the same person or subject are shown in Figs. 3.20, 3.21, 3.22, 3.23, 3.24 and 3.25, where each figure represents a hand image acquired by applying a specific spectrum of light [9].

In this study, right hand images of S460 from the CASIAMS database are employed, because the spectrum wavelength 460nm contains the FTs as stated in [149, 150]. Furthermore, it is a good opportunity to study the specifications of the FTs under a spectral light. The wavelength 460nm represents a visible blue spectrum, where its wavelength value is between $492\text{nm} - 455\text{nm}$ and this range is for the blue colour spectrum as given in Table 3.1 [151].

Table 3.1 shows the offer wavelength of the multi-spectral lighting types. The applied lights: 630nm and 700nm belong to the red spectrum; 850nm and 900nm are included in the near infra-red spectrum; and the white is the visible white light that combines all the visible spectrum colours.

Figure 3.20: Sample of a right hand image from the CASIAMS database for the wavelength lighting 460nm, where the outer texture of the skin is clarified

Figure 3.21: Sample of a right hand image from the CASIAMS database for the wavelength lighting 630nm, where the inner veins and outer texture of the skin are shown

Figure 3.22: Sample of a right hand image from the CASIAMS database for the wavelength lighting 700nm, where the inner veins and outer texture of the skin are shown

Figure 3.23: Sample of a right hand image from the CASIAMS database for the wavelength lighting 850nm, where the inner veins of the skin are demonstrated

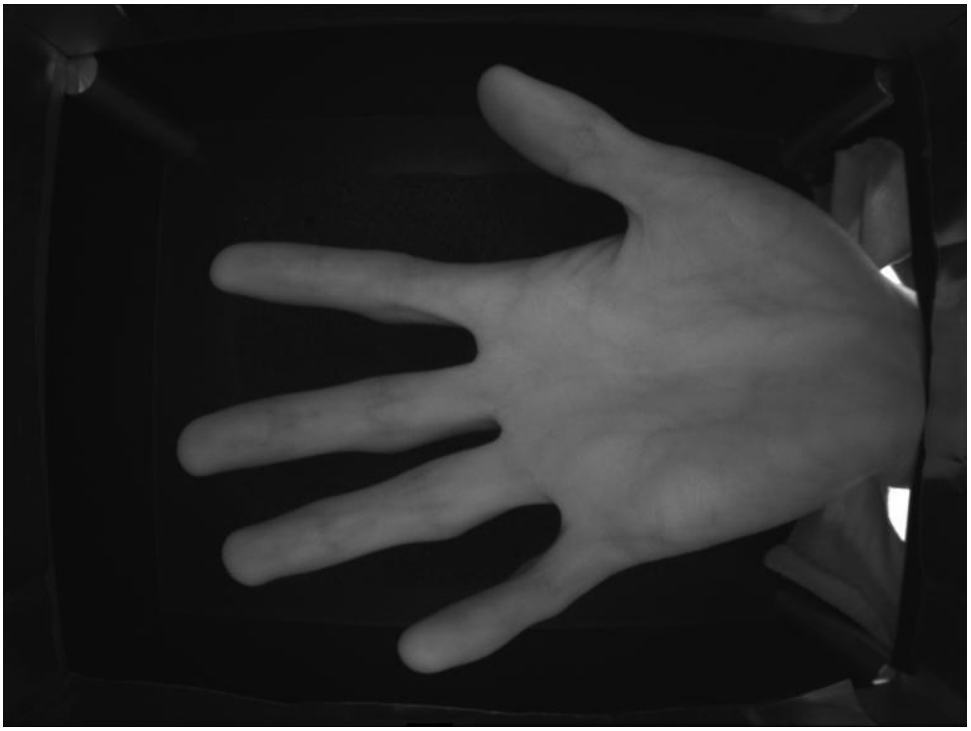


Figure 3.24: Sample of a right hand image from the CASIAMS database for the wavelength lighting 940nm, where the inner veins of the skin are demonstrated

Figure 3.25: Sample of a right hand image from the CASIAMS database for the white lighting, where the outer texture of the skin is clarified

Table 3.1: The wavelengths and frequencies of the visible spectrum colours

Colour	**Wavelength (nm)**	**Frequency (10^{12} Hz)**
Red	$780 - 622$	$384 - 482$
Orange	$622 - 597$	$482 - 503$
Yellow	$597 - 577$	$503 - 520$
Green	$577 - 492$	$520 - 610$
Blue	$492 - 455$	$610 - 659$
Violet	$455 - 390$	$659 - 769$

3.2.4 The Hong Kong Polytechnic University Finger Image Database (Version 1.0)

The Hong Kong Polytechnic University Finger Image (PolyUFI) database (Version 1.0) is probably the first database that considers the FTs of fingers. However, the principle idea beyond establishing this database is to collect a wide range of Finger Veins (FV) images. An acquisition device to capture both the FT and FV images was designed. The architecture of this device is illustrated in Fig. 3.26. This image capturing device was used in The Hong Kong Polytechnic University campus to accumulate the database. The time period of acquiring the data was generally between April-2009 to March-2010. Both genders male and female, were considered in this project, and 156 people provided their finger images. Each image is of a bitmap type and the total number of finger images was 6264. The ages of the participants were under 30 years, which were appropriately 93% of overall subjects. Two sessions were organized to simultaneously capture the FV and FT images with an average interval equal 66.8 days (minimum one month and maximum more than six months). Twelve images were acquired in each session (6 images for the FV and 6 images for the FTs) from only the index and middle fingers respectively. So, in each session 24 images were collected for both fingers. The left hands were only used in this database. Samples of FT and FV images are given in Figs. 3.27 and 3.28 respectively [92].

Many drawbacks can be investigated in this database. First of all, it contains very small regions of FTs. Secondly, just two fingers were employed in this database. Thirdly, part of fingerprints are captured together with a small part of FTs. Fourthly, just the upper knuckles are fully present. Fifthly, as mentioned this database was established fundamentally for FV and in [10], where the database was reported, a fusion method has been mainly exploited depending on the vein

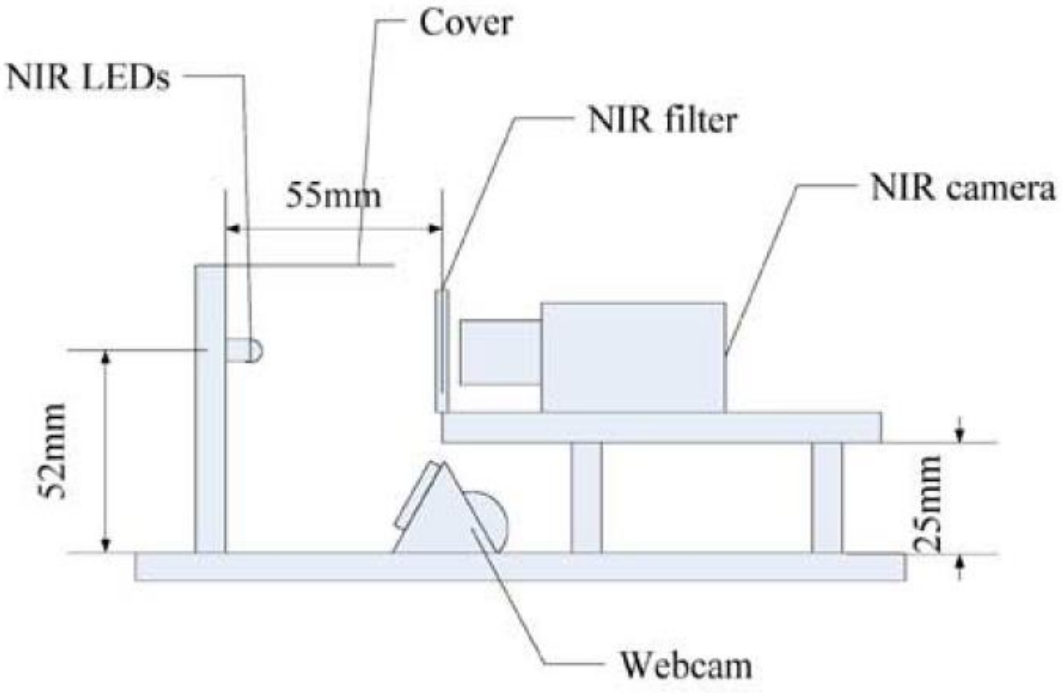


Figure 3.26: Representation of the FV and FT image capturing device [10]

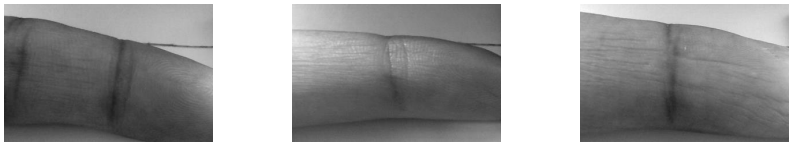


Figure 3.27: Samples of FT images from the PolyUFI. Each FT image belongs to a subject

Figure 3.28: Samples of FV images from the PolyUFI. Each FV image belongs to a corresponding subject to FT image in the previous figure

patterns. Therefore, a comprehensive study cannot be established for the FTs by using this database. It is neglected in the practical contributions of this work.

Part of PolyUFI database can be found in [152], where it is known as The Hong Kong Polytechnic University Low Resolution Fingerprint (PolyULRF) Database (Version 1.0). It composes of only the FT images of the index finger. This

database was investigated in [153] as it consists of very low resolution FT images ($\sim$ 50 dpi) and obtained promising results in terms of identification. Again, only a small part of the index finger image was employed, so, one can argue that this is not enough to achieve best performance or represent a comprehensive study.

3.3 Performance Measures

As mentioned, three databases have been employed in this work: PolyU3D2D database [67], IIT Delhi Database [65] [66] and S460 from the CASIAMS palmprint database [9]. A total of 8850 finger images acquired from 1770 hand images has been employed for the PolyU3D2D database, where each person has contributed 10 images. For the IIT Delhi Database, 4440 finger images have been utilized from 888 hand images. Each person has participated with 6 images. Furthermore, 3000 finger images extracted from 600 hand images within the 460nm wavelength part of the CASIAMS database have been used. Also, each person has contributed 6 images. The reason for using the spectral number 460nm is because it consists of the texture patterns according to [149, 150].

The following process has been implemented on all the 2D right hand images in all the three databases: 5 samples for each person or subject have been utilized in the training phase following [25, 58]; the remaining samples have been used in the testing stage and all the five finger images (thumb, index, middle, ring and little) have been considered in this thesis.

Subsequently, Region of Interests (ROIs) of the five finger images have been extracted and image resizing is equally normalized to cover their textures in the later processing. In this study, the size of each ROI is empirically selected equal to 30×150 after any applied feature extraction function. Then, the resulting image has been partitioned into non-overlapping blocks of equal sizes. Statistical calculations have been applied and a one dimension vector has been prepared for each ROI finger image. So, five feature extractions have been used as the input vector to any employed classifier. Two multi-classifiers have been evaluated, these are: the Feature Level Fusion with Probabilistic Neural Network (FLFPNN) and Finger Contribution Fusion Neural Network (FCFNN). Both of them will be illustrated in Chapter 6. The output of each classifier is aimed to generate the verification decision, where each neuron in this layer refers to a person. The

number of output values of the applied classifiers is equal to the number of the output neurons and this is the same as the number of subjects. This equalled 177 neurons in the PolyU3D2D database, 148 neurons in the IIT database and 100 neurons in the CASIAMS (S460) database, where these are exactly equal to the number of people who provided their images in each database. The output decision values are set as logic '1' for the winner and '0's in all other nodes and there should always be a winner. This procedure is known as the winner-takes-all rule. The number of the failed samples has been counted and recorded for each feature extraction method. To illustrate, when any class is wrongly set to '0' another class will be activated to '1' at the same time. Thus, it is always possible to calculate a false rejection state equal to a false acceptance state in the output of the used multi-classifiers. Consequently, the False Rejection Rate (FRR) is computed equal to the False Acceptance Rate (FAR) and this principally represents the Equal Error Rate (EER) value.

Due to the limitation of calculating the FAR and FRR values in the winner-takes-all rule for the employed multi-classifiers, a novel method to extract FAR and FRR values and establish a relationship between them is proposed. This method is illustrated in Appendix 1 of the thesis.

3.4 Summary

In this chapter, the available FT databases were reviewed. In particular, three databases have been utilized each with a substantial number of finger images: the PolyU3D2D database contains 8850 finger images; the IIT Delhi Database involves 4440 finger images and the CASIAMS (S460) includes 3000 finger images extracted from 1770 hand images; 888 hand images and 600 hand images respectively.

Other FT databases were highlighted, namely the PolyU3D2DV2 and PolyUFI. Both of these databases have not been considered in the later studies in this thesis because of their limitations. That is, the PolyU3D2DV2 has not been completely provided and the PolyUFI contains part of FTs for just two fingers.

In terms of pre-processing measurements, normalizations of the input and output vectors of the applied multi-classifiers have been fairly calculated. All the databases have been divided into two parts; one part is used in the training phase and the other in the testing phase. This will ensure that the neural networks will test new

finger images which have not given before. The assigned neuron will output a logic '1', while '0s' will appear at the other output neurons. The EER value, which is the main parameter of measuring the verification performance, has been calculated.
The next three chapters will explain the contributions of this thesis by using the employed databases and utilize the EER performance measurement mentioned in this chapter.

Chapter 4

Finger Segmentation and Extracting the Region of Interest

4.1 Introduction

The first stage of personal recognition based on Finger Textures (FTs) is segmenting the fingers then extracting the Region of Interest (ROI). All the following stages will depend on how FTs can be successfully collected. Therefore, any failure in this stage may certainly lead to a wrong recognition decision. Two approaches are suggested to address the Finger Segmentation issue:

- A Robust Finger Segmentation (RFS) method, and
- An Adaptive and Robust Finger Segmentation (ARFS) method.

The RFS method is described in this chapter, where each finger is considered as an object. In particular, a robust approach is introduced to extract the five finger images (thumb, index, middle, ring and little) based on an object detection method. An ARFS method is also proposed to address the problem of variation in the alignment of the hand. As such, it can be adapted to different hand alignments such as rotations and translations. A scanning line is suggested to detect the hand position and determine the main specifications of the fingers. Furthermore, an adaptive threshold and adaptive rotation step are exploited.
In the case of extracting the ROIs, a useful method based on an adaptive inner rectangle which has been suggested in [58] is employed. That is, the largest inner rectangle can be obtained adaptively to collect as many features as possible. This

approach has been cited to be the best method in collecting the FTs from the different sizes of fingers, more features are thereby collected from each finger [25]. Then a fixed ROI procedure is applied to all finger surfaces.

The rest of the chapter is organized as follows: Section 2 considers identifying the main points of the fingers. Section 3 explains the proposed segmentation approaches. Section 4 illustrates extracting the ROI. Section 5 demonstrates the results and discussions. Finally, Section 6 summarizes the chapter.

4.2 Main Points of the Fingers

It is important to specify firstly the main finger points (tips and valleys). Secondly, other points can be determined from the boundaries, such as the symmetric or reference point before the index finger and the symmetric or reference point after the little finger. The hand image with the essential points is demonstrated in Fig. 4.1. In this figure $\{T_0, T_1, T_2, T_3$ and $T_4\}$ represent the tips for the thumb, index, middle, ring and little fingers respectively. Whereas, the points $\{V_0, V_1, V_2$ and $V_3\}$ are given for the valleys between the fingers as follows: V_0 is the valley point between the thumb and index, V_1 is the valley point between the index and middle, V_2 is the valley point between the middle and ring, and V_3 is the valley point between the ring and little. The symmetric or reference point before the thumb is marked as R_0, the symmetric point before the index finger is marked as R_1 and the symmetric point after the little finger is marked as R_2. The set points $\{R_0, R_1$ and $R_2\}$ can be specified in such a way that the length of the straight line T_0-V_0 is equal to the length of the straight line T_0-R_0; the length of the straight line T_1-V_1 is equal to the length of the straight line T_1-R_1 and the length of the straight line T_4-V_3 equals the length of the straight line T_4-R_2. A detailed description of this method is found in [117, 118].

Additional points have been determined in the finger base to specify the finger direction. The finger base is defined as the line between the two valley points or between the valley and symmetric points around the finger. See the following equations below [25]:

$$VR_{3\_}x = \frac{VR_{1\_}x + VR_{2\_}x}{2} \tag{4.1}$$

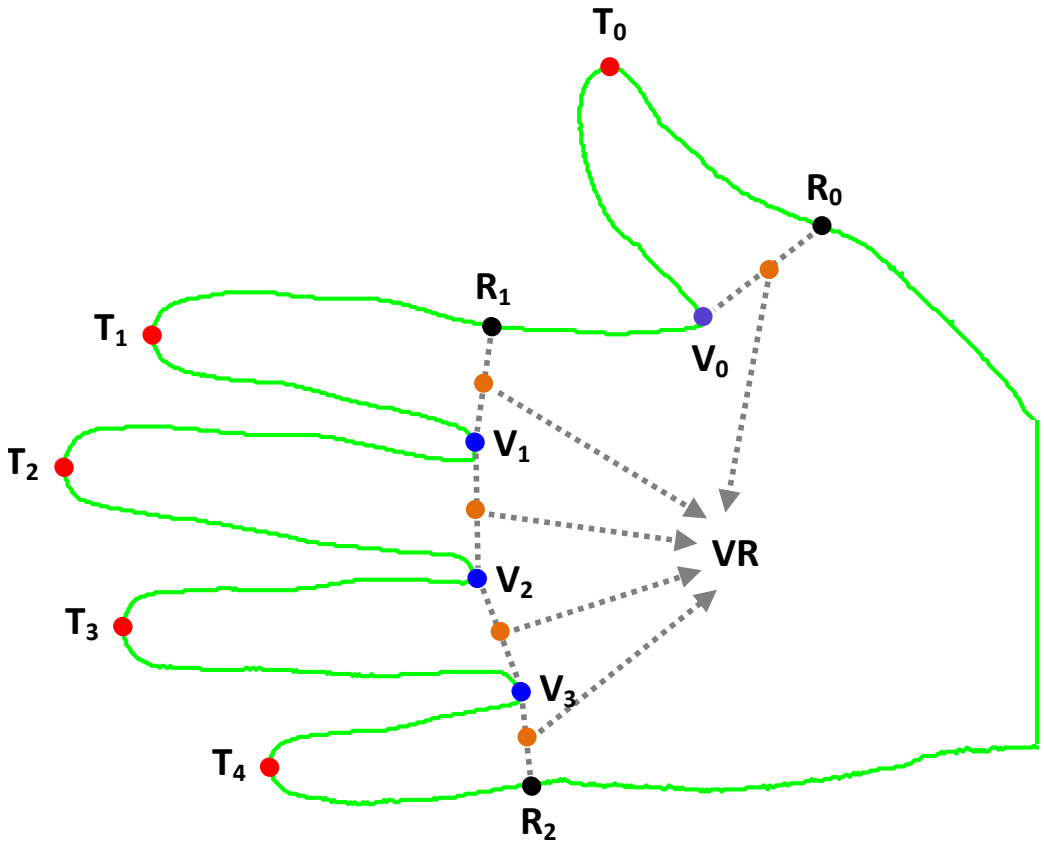


Figure 4.1: A hand image boundary with the essential points: tips, valleys and symmetric points where $\{T_0, T_1, T_2, T_3$ and $T_4\}$ represent the tips, $\{V_0, V_1, V_2$ and $V_3\}$ represent the valleys and $\{R_0, R_1$ and $R_2\}$ represent the symmetric

$$VR_3\_y = \frac{VR_1\_y + VR_2\_y}{2} \tag{4.2}$$

where VR_3 represents the centre point of the base line, VR_1 and VR_2 represent a valley and a symmetric point or two valley points, and x and y represent the x_axis or y_axis coordinate respectively.

These equations could be repeated between each two valley points, or even between a valley and a symmetric point to specify the new point in the centre base of a finger length. The finger length is defined as the length between the tip and the centre point of the finger base line [117, 118].

The ground truth of the finger segmentation basically depends on the finger base and finger tip [60, 129, 154, 155]. However, before extracting the finger image in this thesis additional pixels have been included to cover the full patterns of the lower

knuckles, which have important features as it has been confirmed in [25]. The length of these pixels have been empirically determined for each database.

4.3 Proposed Finger Segmentation Approaches

4.3.1 Robust Finger Segmentation Method

The extraction of the FT pattern from a hand image is not a straightforward procedure. Therefore, different image processing algorithms are employed to extract the five fingers (thumb, index, middle, ring and little finger) from a hand image. In this chapter, the RFS segmentation method is proposed to explore the fingers as objects. Then, the finger segmentations can be carried out after specifying the main finger points, which are mentioned in the previous Section 4.2. The image preprocessing steps begin by reading the colour image and subsequently converting it to an 8-bit grayscale image denoted as $I(x, y) : Z^2[0, 255]$. An example converted grayscale hand image is given in Fig. 4.2.
Henceforth, it is translated to a binary image by using a threshold τ. The binarization process is represented by Equation (4.3) [156, 157]:

$$B(x,y) = \begin{cases} 1 & \text{if} \quad I(x,y) > \tau \\ 0 & \text{if} \quad I(x,y) \leq \tau \end{cases} \tag{4.3}$$

where $B(x, y)$ is the resulting binary image, $I(x, y)$ is the grayscale image and τ is the threshold. It has been observed that it is better for the value τ to be adaptive as this will preserve the hand image area rather than the finger objects. For The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67] and Spectral 460nm (S460) from the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9] the threshold value is adapted between 0.1 and 0.17, and for the red channel of the Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66] the threshold value is adapting between about 0.21 and 0.3. To justify choosing the threshold value for a grayscale hand image in any database, several steps can be utilized as follows:

1. Initializing the threshold τ to a small value.

2. Applying the binarization process.

3. Carrying out the operations of detecting finger objects.

4. If the finger objects are incorrectly specified, the τ value must be slightly increased. Then, go to step 2 and repeat all the above operations.

5. If the finger objects are correctly determined, then the segmentation operations can be continued.

There will still be, however, some binary noise outside the hand region as shown in Fig. 4.3. So, the following steps can be used to remove them [158]:

1. Specifying the white areas.

2. Calculating the size of each area.

3. Deleting all white areas except the largest one which represents the hand.

The resulting image is then denoted as $B(x, y)$, see Fig 4.4. Thus, the complement will be defined as $\bar{B}(x, y) : Z^2 \{1, 0\}$. Nevertheless, there may still be some unexpected noise connected to the hand image as shown in Fig. 4.5.
To overcome the effects of this remaining noise the 'majority' filter morphological operation is applied to the complement of the binary image. This filter can be represented according to the following equations [159]:

$$\boldsymbol{B}_{\text{majority}}\mathbf{(q)} = \mathbf{MAJ[SE.B(q)]} \tag{4.4}$$

$$\mathbf{MAJ}(\mathbf{z}) = \begin{cases} 0; \text{ if a majority of the elements of } \mathbf{z} = 0 \\ 1; \text{ if a majority of the elements of } \mathbf{z} = 1 \end{cases} \tag{4.5}$$

where **B** represents the binary image, **q** represents the image coordinate, **MAJ** represents the majority process, **SE.B(q)** represents the blocking set, the **SE** represents the structuring element which is a kernel matrix with a fixed number of pixels equal to $2M + 1$ and $\mathbf{z} = [Z_1, Z_2, ..., Z_{2M+1}]$ refers to arbitrary binary pixels. The **SE** size is equal to 3×3 and $2M + 1$ is equal to 9.
In other words, it is performed by converting the black pixels with the majority

white neighbourhood to logical one [160, 161]. The complement image after the 'majority' morphological filter is given in Fig. 4.6.

In the case of specifying the main point of the thumb, another scanning operation is performed for the complement image $\bar{B}(x, y)$ until two areas are detected: a small area, which represents the thumb, and a large area for the rest of the fingers with a part of the hand as shown in Fig. 4.7. The tip point of the thumb can be determined as the furthest top side point from the object. On the other hand, the valley point can be specified simply by assigning the separating point between the two objects.

Another scan can then be executed to verify the four finger objects (index, middle, ring and little). Empirically, this was performed by cropping the hand area to two fifths of its total length. Each object could be assigned by a fixed pixel value or one colour. So, the four finger objects could be represented by four different colours with a black background as shown in Fig. 4.8. Hereafter, the tip points will be assigned where each point can be defined as the furthest left point from the object, see Fig. 4.9.

Similarly, the background is converted into colour objects to specify easily the

Figure 4.2: Original grayscale hand image

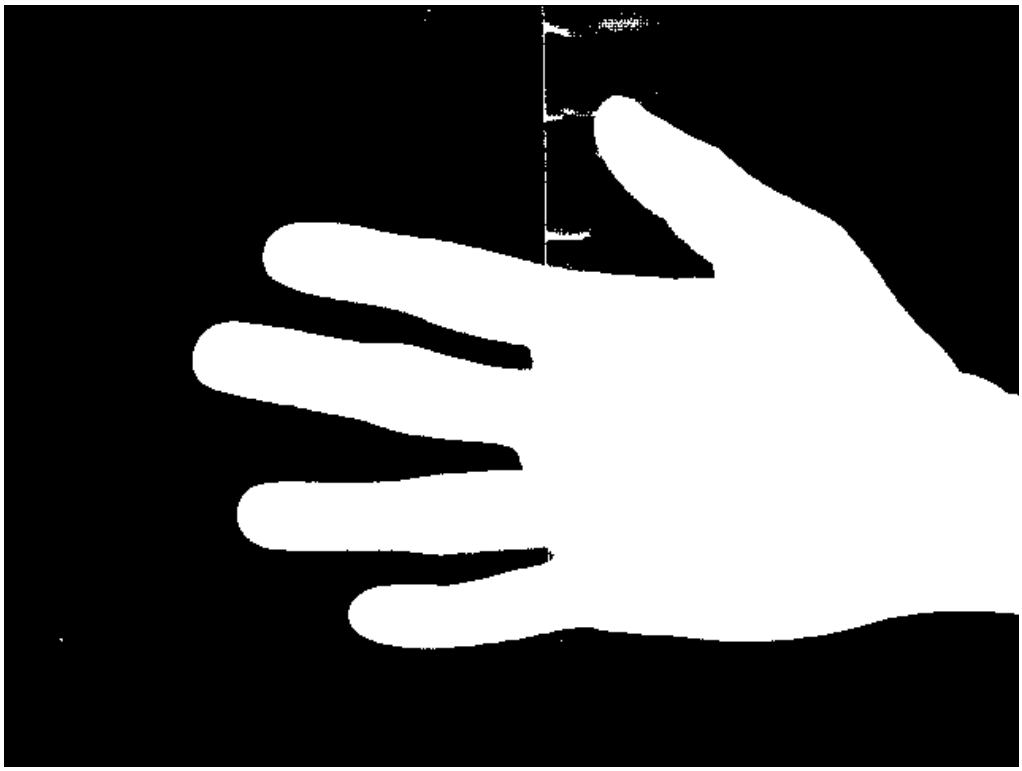


Figure 4.3: Converted binary image from the grayscale image

Figure 4.4: Binary image after deleting the small white noise areas

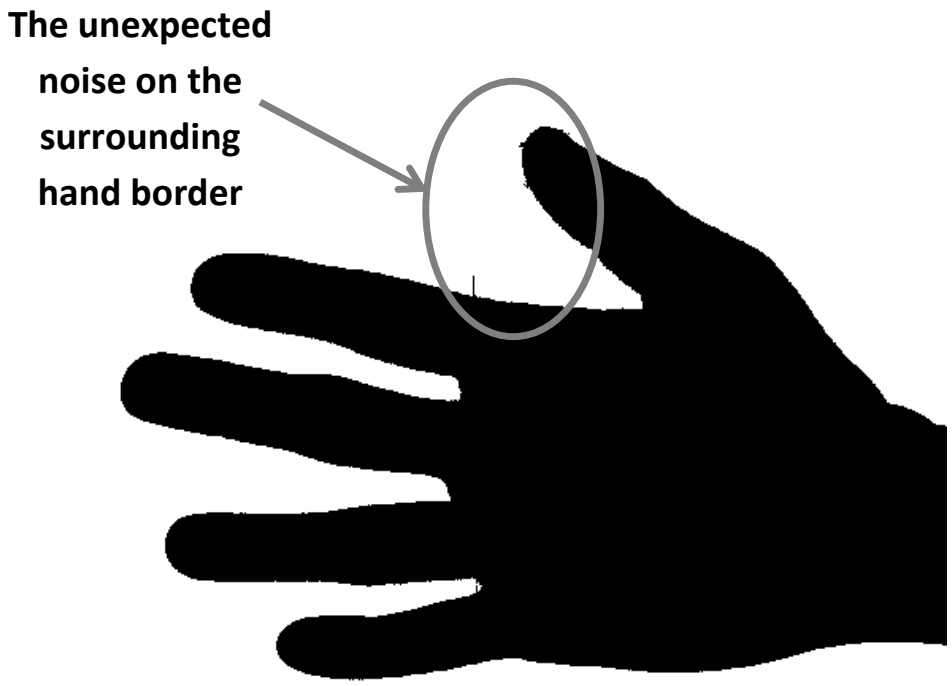


Figure 4.5: Image complement of the previous binary image

Figure 4.6: Image complement after the 'majority' morphological filter

Figure 4.7: Thumb object with the tip and valley points

Figure 4.8: Four finger objects each object assigned by a specific integer value

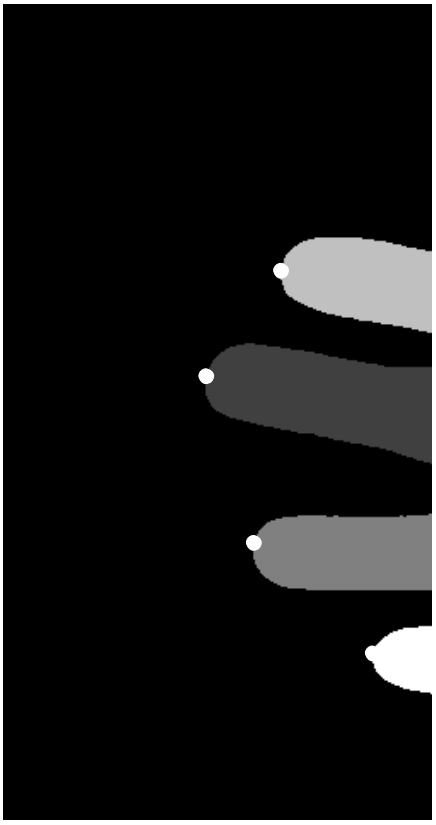


Figure 4.9: Four finger objects with the tip points

Figure 4.10: Background objects assigned by specific colours with the valley points

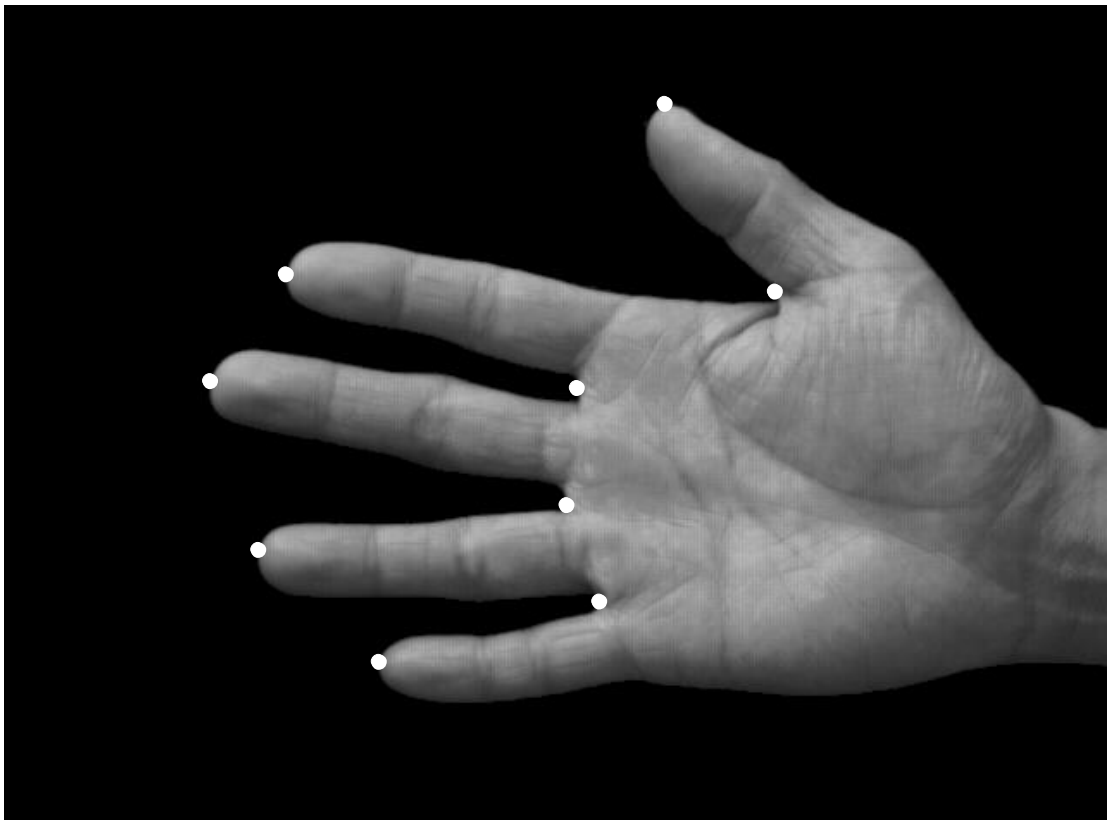


Figure 4.11: Original hand image with the tips and valleys points

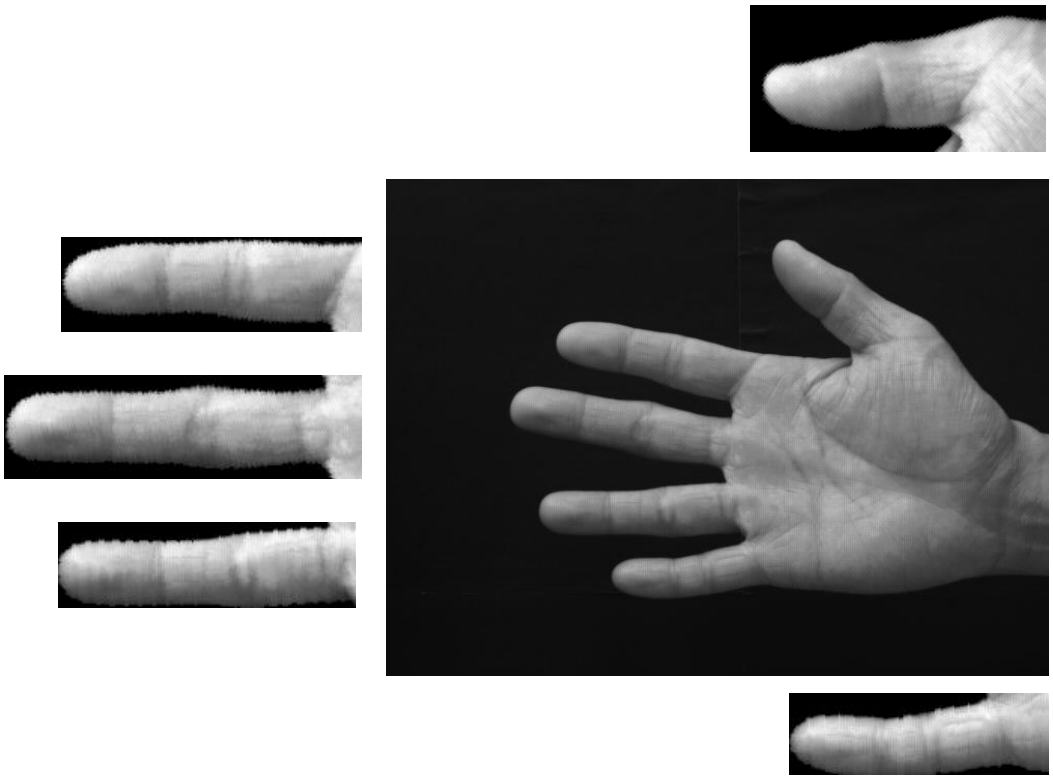


Figure 4.12: An example of a hand image with its segmented fingers

valley points. Each tip point is extended to be a border between the two objects as given in Fig. 4.10. From this point the valleys have been assigned between the fingers, where each valley point can be denoted as the furthest right point from the object. A hand image with all tips and valley points is given in Fig. 4.11.

Henceforth, after specifying the main points of the fingers (the tips and valleys) and considering the symmetry in the opposite side of the valleys around the terminal fingers (little, index or thumb), the centre points of the base line can be calculated according to Equations (4.1) and (4.2). Then, the original hand image will be combined with the pure black background as this is more robust than using the original background. See Equation (4.6):

$$I_{new}(x,y) = \begin{cases} 0 & \text{if} \quad B(x,y) = 0 \\ I(x,y) & \text{if} \quad B(x,y) = 1 \end{cases} \tag{4.6}$$

Then, this image needs to be rotated. The best rotation angle to be established is based on the finger length. Consequently, the finger direction angle in degrees can be calculated according to the following equation:

$$\gamma = r \ (tan^{-1} \frac{\mid Tip\_y - VR_3\_y \mid}{\mid Tip\_x - VR_3\_x \mid} \times 180/\pi) \tag{4.7}$$

where γ is the orientation angle of a finger and r is for the sign of the angle (1 or -1).

Equation (4.7) will be used for each finger in order to determine its direction to be extracted in a separate image. An example of a hand image and segmented fingers is shown in Fig. 4.12.

The proposed finger extraction is appropriate for contactless (peg-free) hand images, where it can deal efficiently with the translation and scaling of the hand images. In addition, it maintains the finger images rather than the hand image.

4.3.2 Adaptive and Robust Finger Segmentation Method

The ARFS method has been inspired from the RFS method and it is suggested to address the problems of different hand alignments. In this section, the full process of FT segmentation will be described by using the IIT Delhi database [65]. The

reason for focusing on this database is because of its specifications, as it involves unrestricted hand images that have various orientations, scalings and translations [66, 148]. However, the method is capable of working with other databases.

First of all, the hand images of the IIT Delhi database have a small difference over the hand images of the other two databases, that is, they are rotated by 90° in the clockwise direction. Simply, this has been solved by rotating any hand image from this database by 90° in the counterclockwise direction at the beginning of the segmentation process. This will allow similar operations and calculations to be used for any input hand image.

Consequently, the coloured hand image has been converted to the grayscale level, but the red channel from the coloured image is still required for the next process (binarization). Practically, it has been noticed that applying the threshold to the red channel is more effective than applying it to the blue or green channels. This is due to the skin colour on inner surface of the hand, where it is commonly related to the red spectrum colour, see Fig. 4.13.

Experimentally, utilizing an adaptive threshold, which refers to adaptively changing the threshold as described in Subsection 4.3.1, has been found to give better results than using a fixed or Otsu threshold. The binarization process is represented by Equation (4.8) [156, 157]:

$$B(x,y) = \begin{cases} 1 & \text{if} \quad I_r(x,y) > \tau \\ 0 & \text{if} \quad I_r(x,y) \leq \tau \end{cases} \tag{4.8}$$

where $B(x,y)$ is the resulting binary image, $I_r(x,y)$ is the separated red channel image from the source coloured image and τ is the threshold.

Then, similar operations of preserving the hand image by removing the small white noises and employing the 'majority' filter which have been described in Subsection (4.3.1) are applied in the ARFS method.

The next important step is finding the hand specifications which involves: the four fingers and the anchor point. The anchor point will be the lower valley point between the thumb and index finger, as long as the main fingers are located in the reference or normal direction. The key idea of this stage is proposing a scanning line to recognize the hand attributes. The scanning line has the following facilities: it can move to any part of the figure according to the scanning requirements; it

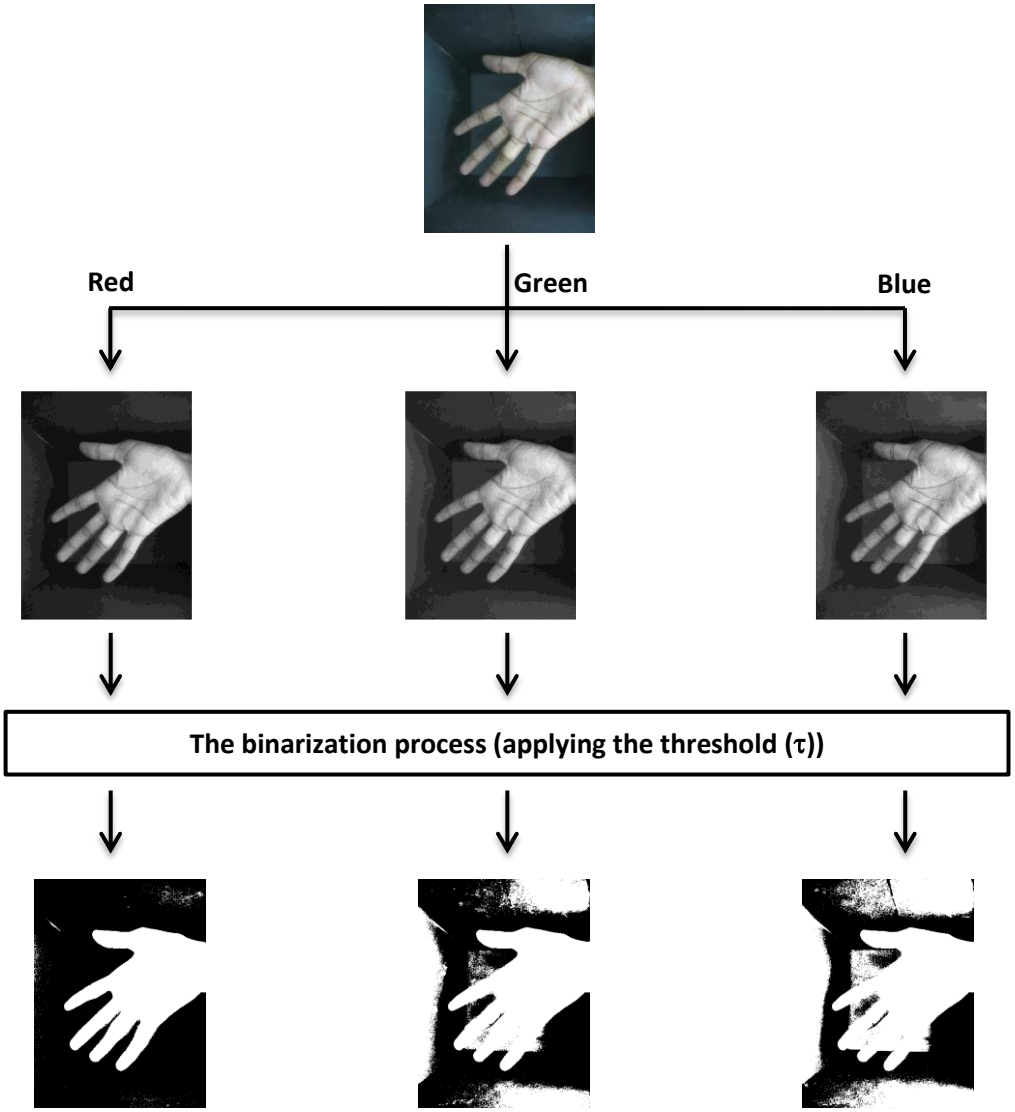


Figure 4.13: Applying a binarization threshold to the red, green and blue channels of a coloured image

consists of a set of points, so, it has the ability to detect specific points located in narrow places; it can efficiently detect separated objects and it analyses the correct direction of the hand.

Moreover, an adaptive rotation has been implemented in order to locate the hand image in an appropriate position. The adaptive rotation is denoted as adaptively adjusting the rotation of the hand position angle. It is worth mentioning that it is not important for the adaptive rotation to be implemented with small steps by changing a small angle, instead, a suitable step angle can be used to reduce computation time.

There are two styles of rotation: 'loose' and 'crop' [162]. The first rotating style

'loose' has been chosen to be used instead of the second style 'crop'. This is because the 'loose' style preserves the image contents and adapts the image size accordingly. Whereas, the second style 'crop' eliminates the image according to the original image size, which might cause some parts of the image to be removed. The new coordinate of the rotated images to the specific angle θ is denoted as (x', y'). Different specifications can be detected by the scanning line during the adaptive rotation as given in Figs. 4.14, 4.15, 4.16 and 4.17. However, the main attributes are the four longest fingers with the anchor point. Consequently, the scanning line can cause a significant number of scans to detect all the other main points (finger tips and valleys).

The grayscale hand image is combined with the pure black background. So, Equation (4.9), which is the modified version of Equation (4.6), can be used after rotating the grayscale image to the right direction:

$$I_{new}(x', y') = \begin{cases} 0 & \text{if} \quad B(x', y') = 0 \\ I(x', y') & \text{if} \quad B(x', y') = 1 \end{cases} \tag{4.9}$$

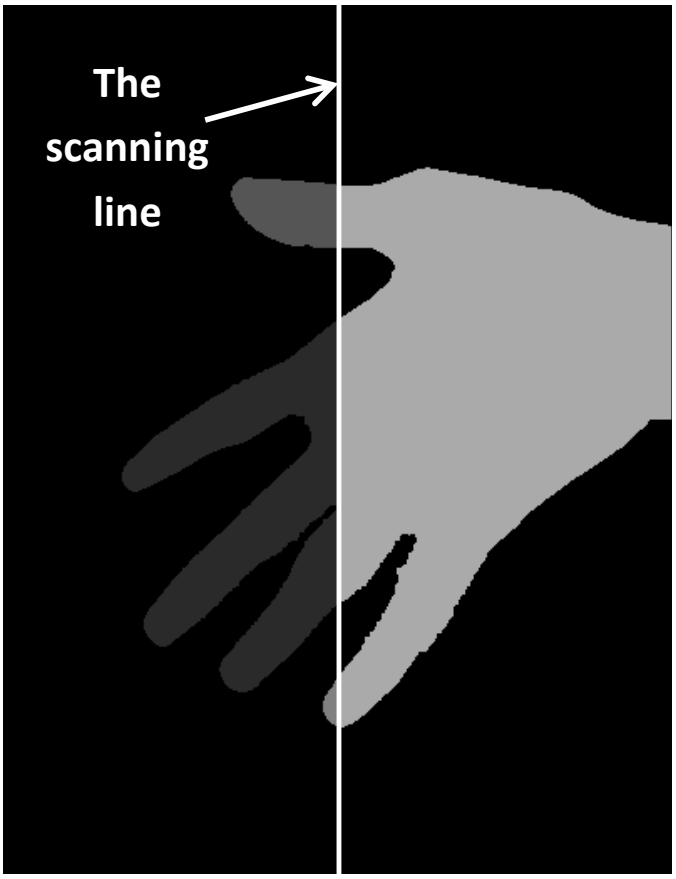


Figure 4.14: Wrong three objects each with an assigned colour

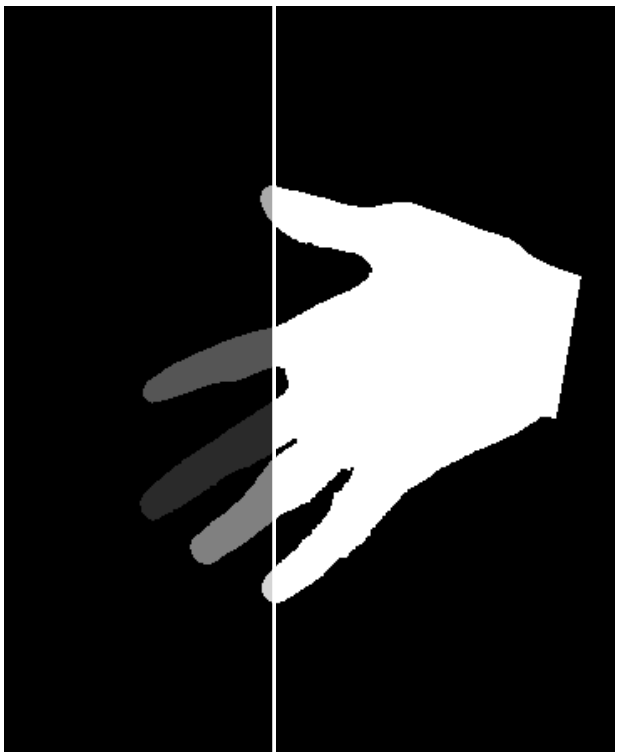


Figure 4.15: Wrong five objects each with an assigned colour

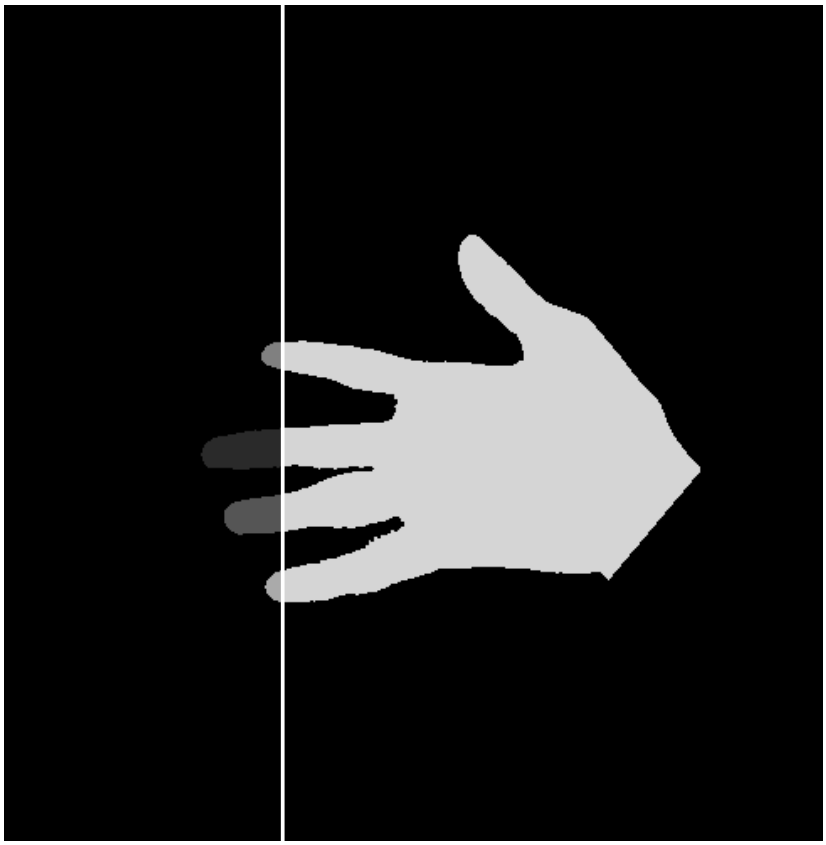


Figure 4.16: True four objects represent the four fingers

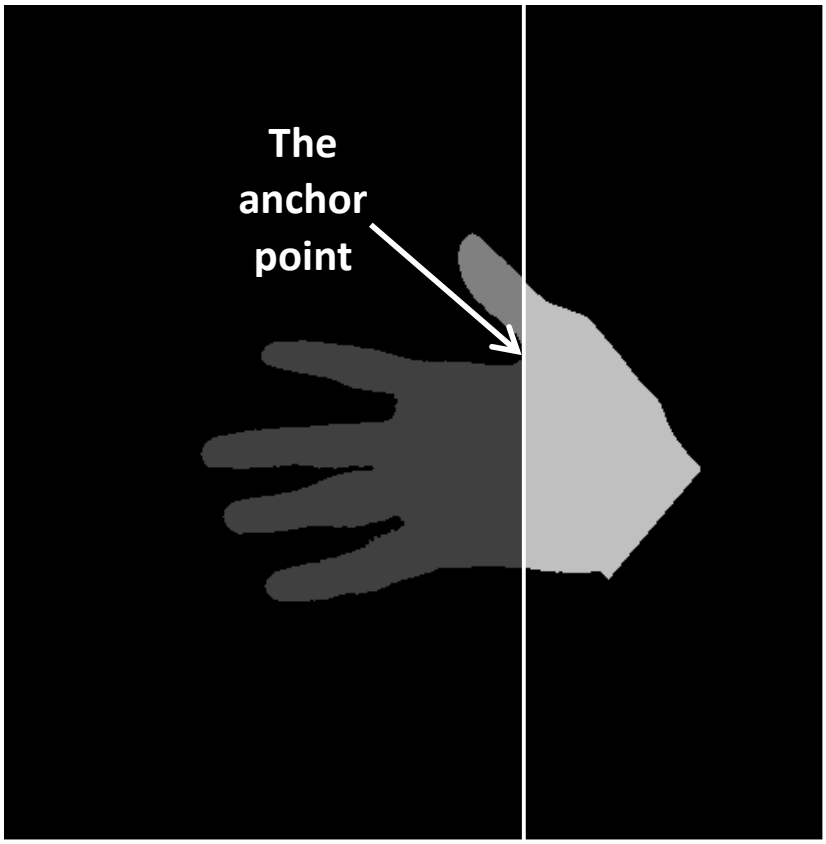


Figure 4.17: The anchor point (the valley point between the thumb and index finger)

Figure 4.18: Sample of a hand image with all tips and valley points

where $I_{new}(x', y')$ is the new grayscale image containing the hand with the black background and $I(x', y')$ is the original grayscale image. A sample of a hand image with its tips and valleys is given in Fig. 4.18.

Symmetric points can also be calculated, as mentioned in Section 4.2, to describe the following locations from the top: before the thumb finger, before the index finger and after the little finger. The same operations which have been performed in the RFS method are considered here to determine the finger orientation. Then, the finger orientation angle can be computed based on the finger length direction according to Equation (4.7). This process should be repeated for each finger.

It is worth illustrating that the resulting images of the finger segmentations by using the ARFS are similar to the resulting segmented finger images of the RFS method. This is because the ARFS is derived from the RFS, so, all the finger points specified by both methods are the same. Therefore, the extracted ROIs are also the same.

4.4 Extracting the Region of Interest

In the case of the ROI, a useful method has been utilized based on the adaptive inner rectangle. This method has been reported in [58], where an adaptive inner rectangle was applied to extract the ROI for just the four fingers (index, middle, ring and little). However, in this publication the lower or third inner knuckle was not considered and thus important features of the FTs are avoided. This knuckle contains important textures according to [25]. A further modified model for the adaptive ROI rectangle is applied here, where more features can be collected for each finger from the ROI areas. All the five fingers including the thumb are considered in this study. The lower knuckles are employed according to [25]. This method is found to be more efficient than extracting the ROIs in [117, 118] as they considered that the ROIs have specific ratios between the width and length of the fingers. It is believed that there is no universal fixed proportional ratio between a finger's length and width. Furthermore, the proposed ROI extraction approach in [44, 45] partially includes the lower knuckle and over covers the finger width, where parts of the background are collected in the ROIs because of the finger geometry.

After specifying the ROIs, a fixed resize has been applied to each finger image in order to normalize them into fixed sized vectors. Empirically, the normalization resize is considered equal to 30×150 after any updated Local Binary Pattern (LBP)

type. An example of segmented finger images with their extracted ROIs according to the adaptive inner rectangle method are shown in Fig. 4.19.

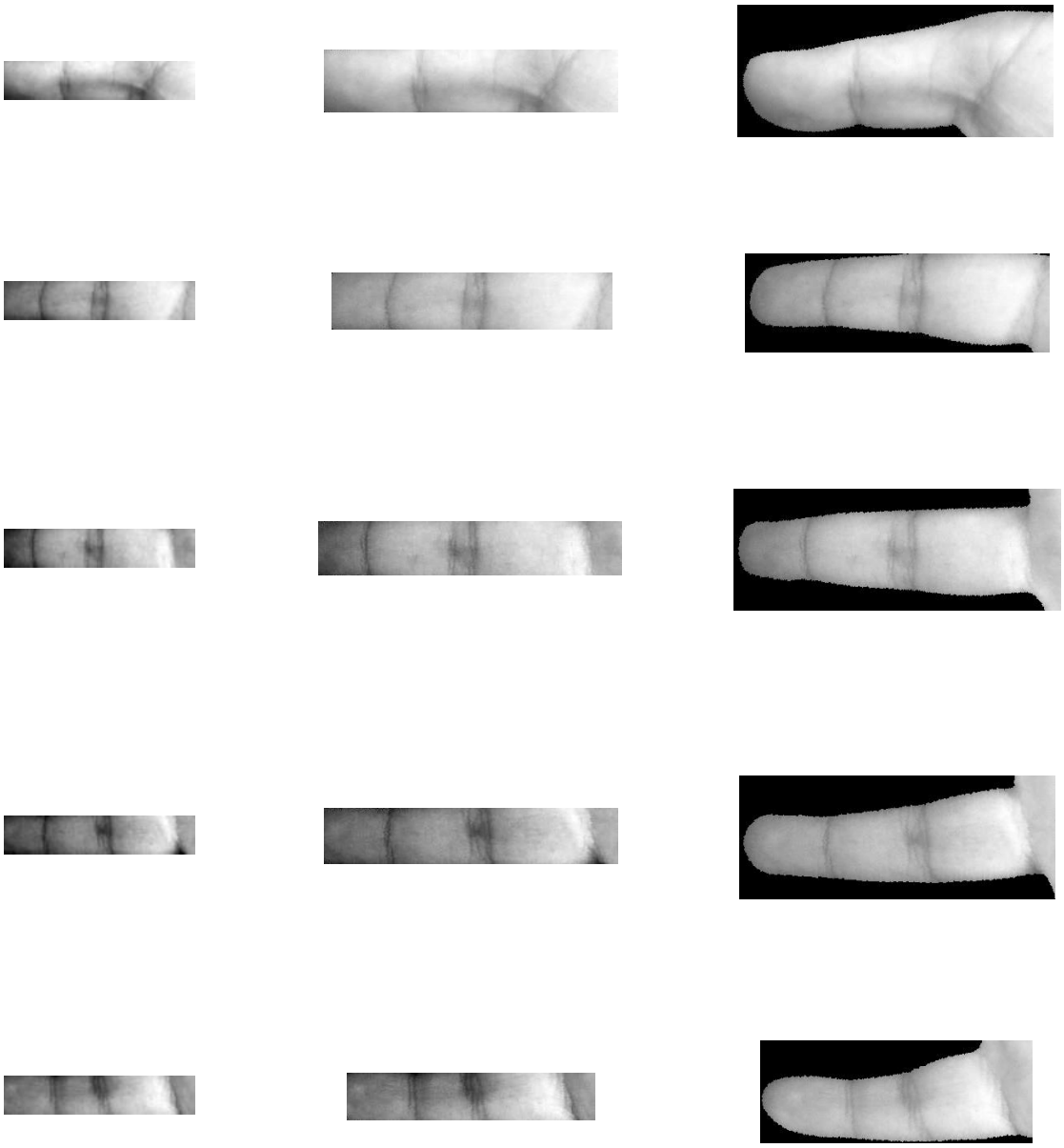


Figure 4.19: Segmented finger images with their extracted ROIs according to the adaptive inner rectangle method. From the top: the first row is for the thumb; the second row is for the index finger; the third row is for the middle finger; the fourth row is for the ring finger; the fifth row is for the little finger. From the right: the first column is for the segmented finger images; the second column is for the extracted ROIs according to the adaptive inner rectangle method; the third column is for the fixed resize of the ROI images

4.5 Results and Comparisons

A comparison has been established between the two finger segmentation methods (RFS and ARFS) and other prior work. This comparison is illustrated in Table 4.1, where a summary of related publications stated in Chapter 2:Section 2.3.1 is shown.

Various methods were compared with the proposed approaches as demonstrated in Table 4.1, where it shows that the proposed ARFS approach can carry out the various degrees of translations, scalings and orientations. Whereas, other methods can just recognize slight rotation and translation movements and various scaling sizes as in [25, 58]. Other finger segmentation approaches were designed to hand images located in specific positions such as [117, 118], where a scanner device was used to collect the hand images. In this case, just small translations could be detected, while, the orientations and scaling movements were not allowed. According to [44] the projectile approach can recognise small movements with a fault tolerance for the translations and orientations, but it could manage the large translations. In the case of orientations, the projectile method can address reasonable large orientations as the authors considered that big orientations are unnatural alignments for the users to their acquisition design. Different scaling sizes can be managed by the projectile method. Same approach was presented in [45] but only for four fingers.

It is worth highlighting that one of the main differences between the RFS and ARFS is the adaptation of investigating the finger objects. The RFS method explores the four finger objects by determining the beginning of the hand area. This is calculated directly by two fifths of the total length of the hand area as shown in Figs. 4.8 and 4.9. That is why the RFS can detect different alignments of the hand translations and scaling. On the other hand, the scanning line strategy in the ARFS model is adaptively moving. By scanning the hand images from the beginning of the four finger objects, it can recognise the orientation alignments of the hand in addition to the translation and scaling alignments.

In the case of resolution, the suggested segmentations are robust for different resolution levels high, low and very low. That is, the IIT Delhi database can be considered as high resolution coloured data, where each hand image has the size $(1200 \times 1600 \times 3)$. The contactless CASIAMS hand images are low resolution, each

with the size (576×768). While, the PolyU3D2D can be considered as a very low resolution coloured images, because its images have the size ($480 \times 640 \times 3$) and the participants were asked to locate their hands far from the acquisition device to a distance approximately equal to 0.7m [67]. On the other hand, specific database resolution was used in other publications, where the segmentation methods were designed for their utilized database. Furthermore, a limited number of fingers was segmented in several studies as in [25, 45, 58, 118], while, a full number of hand fingers was considered in the others. Another point to be mentioned is that the suggested segmentation methods are robust in extracting the fingers after efficiently eliminating the hand image noises by utilizing the adaptive threshold. That is because the fixed threshold works better for high resolution hand images with pure black background as in [117, 118], where their data were acquired from a touch scanner. Furthermore, the Otsu threshold, which is proposed in [119], does not maintain the full hand object rather than the fingers in the images. That is why additional opening morphological operations were required in [58] after the binarization process. While, these operations were not required after using the other types of the threshold like the fixed threshold in [117, 118] and the adaptive threshold in the suggested segmentation approaches.

The suggested finger extraction strategies have fewer points and less complexity, where 4 points have been used in this work including the essential points to detect the finger orientation compared with 9 points established in [58] and 10 points defined in [117, 118]. In contrast, assigning more points is more likely to be used in the high resolution images as in [117, 118]. Also, using the adaptive inner rectangle area suggested in [58] for each finger is already covering the features of the wide finger area.

In the case of extracting the ROIs, a comparison summary of related work for the finger images of right hands is shown in Table 4.2. The majority of the ROI extracting approaches collected just part of the FTs. For instance, the distal phalanx was excluded and the upper knuckle was partially included in [118]. Whereas, they have been considered in all other compared methods. The lower knuckles were partially included in [117, 118] and not included in [58]. This caused some important FT features to be lost. In addition, the thumb is neglected in [25, 45, 58, 118] which could perform enhancements to their final results. In [117] the length of ROI was specified to be five sixths of the length of the finger and the

Table 4.1: Comparison summary between the suggested segmentation methods and related work

Comparison domain	**Proposed RFS**	**Proposed ARFS**	**Ribaric and Fratric** [117]	**Pavesic *et al.* [118]**	**Kanhangad *et al.*[58]**	**Michael *et al.* [44]**	**Michael *et al.* [45]**	**Al-Nima *et al.* [25]***
Detecting hand translations	Small and large translations	Small and large translations	Small translations	Small translations	Small translations	Small (with fault tolerant) and large translations	Small (with fault tolerant) and large translations	Small translations
Detecting hand orientations	Small orientations	Small and large orientations	—	—	Small orientations	Small (with fault tolerant) and reasonable large orientations	Small (with fault tolerant) and reasonable large orientations	Small orientations
Detecting hand scaling	Small and large scaling	Small and large scaling	—	—	Small and large scaling	Small and large scaling	Small and large scaling	Small and large scaling
Database Resolution	High, low and very low resolutions	High, low and very low resolutions	High resolution	High resolution	Very low resolution	Low resolution	Low resolution	Very low resolution
Number of segmented fingers	Five fingers	Five fingers	Five fingers	Four fingers	Four fingers	Five fingers	Four fingers	Four fingers
Binarization threshold	Adaptive threshold	Adaptive threshold	Fixed threshold	Fixed threshold	Otsu threshold	—	—	Adaptive threshold
Number of assigned points	4	4	10	10	9	Changeable	Changeable	4

* This was the first author's paper relating to verifying people by utilizing the FT features. It used the traditional contour method as a basis to segment the four fingers from a hand image.

Table 4.2: Comparison summary between the exploited ROIs extraction method and prior studies

Comparison domain	**Proposed RFS**	**Proposed ARFS**	**Ribaric and Fratric** [117]	**Pavesic *et al.* [118]**	**Kanhangad *et al.*[58]**	**Michael *et al.* [44]**	**Michael *et al.* [45]**	**Al-Nima *et al.* [25]***
Distal Phalanx	Include	Include	Include	Not include	Include	Include	Include	Include
Upper knuckle	Include	Include	Include	Partially include	Include	Include	Include	Include
Lower knuckle	Include	Include	Partially include	Partially include	Not include	Partially include	Partially Include	Include
Finger width	Include	Include	Not always include	Partially include	Include	Over include	Over include	Include
Strategy	Adaptive inner rectangle	Adaptive inner rectangle	Fixed proportional strip	Fixed proportional strip	Adaptive inner rectangle	Projectile	Projectile	Adaptive inner rectangle
Excluded ROI	None	None	None	Thumb	Thumb	None	Thumb	Thumb

* This was the first author's paper relating to verifying people by utilizing the FT features. It employed the adaptive inner rectangle strategy [58] to extract the FT regions from the four fingers. However, the lower knuckles were also included.

ROI's width was assigned to have a fixed ratio with the ROI's length; this ratio was determined empirically for each finger. It can be argued that the ratio between the ROI length-width is not fixed between individuals, that is, the proportional ratio for the length-width of the person who has long and fat fingers does not equal to the proportional ratio of the person who has the same long but thin fingers. Similarly, in [118] the ROI length was determined after dividing the finger length by 1.5 and the ROI width was assigned after dividing the finger width by 2.3. So, in [117] the full finger's width was not always collected. On the other hand, in [58] an adaptive inner ROI rectangle has been suggested to fit in the finger length and width, but in this work the lower knuckles and the thumbs were not included as mentioned before. The ROIs in [44, 45] have a drawback in collecting the widths of the fingers. That is selecting the outer rectangle in the ROI of each finger instead of the inner rectangle, which causes parts of the backgrounds relating to the finger form geometry to be collected. In the contributions of this chapter, a modified model of the adaptive inner rectangle has been employed, where more features have been proposed to be collected for each finger from the ROI areas. Some studies ignored the ROI of the thumb [25, 45, 58, 118], while, it has been considered in the two suggested segmentation approaches.

In addition important features of the lower knuckles are included in the ROI of all fingers and this has decreased the error rate of the FT biometric verification as given in Table 4.3, where a comparison according to the state-of-the-art is given.

Kanhangad *et al.* [58] have employed a Competitive Coding (CompCode) method, as a feature extraction, with the Hamming Distance (HD), as a matching metric between the testing vectors and the templates, to the PolyU3D2D database. High Equal Error Rate (EER) value equal to 6% was benchmarked in this publication. From Table 4.3 it is clear that using more features will increase the successful performance of the verification. This issue has been recorded in different feature extraction methods as in [25] for the four fingers, where a new feature extraction method has been applied named Image Feature Enhancement (IFE). Generally, the EERs after adding the third or lower knuckle are better than the EERs without this important feature such as in the IFE based exponential histogram the EER percentage has been reduced from 5.42% to 4.07%. Similarly, adding the FTs of the thumb will enhance the verification performance compared with using just four fingers.

Table 4.3: The EER results for four fingers without inner knuckles, four fingers with inner knuckles and five fingers with inner knuckles

Method	**Additional factors**	**Database**	**EER (4 fingers without inner knuckles)**	**EER (4 fingers with inner knuckles)**	**EER (5 fingers with inner knuckles)**
CompCode [58]	Hamming distance	PolyU3D2D	6%	—	—
IFE [25]	For flat histogram	PolyU3D2D	5.54%	4.07%	—
	For exponential histogram	PolyU3D2D	5.42%	4.07%	—
	For bell-shaped histogram	PolyU3D2D	12.66%	7.01%	—
Suggested MSALBP	P=8, R=2	PolyU3D2D	1.02%	0.79%	0.68%
	P=16, R=2	IIT Delhi	—	2.03%	1.35%
	P=8, R=2	CASIAMS(S460)	—	5%	2%
Suggested ELLBP	N=17	PolyU3D2D	0.68%	0.45%	0.34%
	N=17	IIT Delhi	—	3.38%	1.35%
	N=7	CASIAMS(S460)	—	2%	0%

All of the aforementioned work in Table 4.3 adopted the same database (the PolyU3D2D). However, in this work two additional databases have been used (IIT Delhi and CASIAMS (S460)). The results in the table are recorded taking into account that the best parameters or factors are considered for the suggested feature extractions, where these parameters achieve the verification percentages that will be confirmed in the next chapter. These results are consistent with the state-of-the-art as there are improvements in EER values after increasing the FT features by including the lower knuckles. In addition, all the five fingers have been applied for the verification evaluation. This reveals that better FT verification performance can be attained after including more features.

4.6 Summary

In this chapter, two finger segmentation approaches were presented. The first approach is called the RFS. This segmentation method represents each finger as an object. It maintains the hand image before the segmentation process is carried out. The second segmentation approach is called the ARFS. This method is inspired from the RFS and it is suggested to address the alignment problems of the hand image.

Detailed explanations about the drawbacks from the prior work were given. A

description of the essential finger points was provided. Then, both segmentation methods were illustrated and demonstrated. An adaptive inner rectangle is explained and employed to extract the ROI for five fingers. Intensive comparisons were established between the proposed segmentation approaches and other reliable publications. Additional comparisons were presented for extracting the ROIs. Finally, verification performance comparisons according to the state-of-the-art were reported and they confirmed that using more FTs would increase the verification performance.

After segmenting the fingers and extracting their ROIs, a feature extraction stage will be carried out in the next chapter.

Chapter 5

Feature Extraction

5.1 Introduction

Feature extraction plays an effective role in any biometric system. So, it is important to consider this aspect in the thesis. Due to the fact that the Finger Texture (FT) has been recently explored as a biometric characteristic, there is no specific feature extraction technique that can efficiently analyse the main FT features (horizontal patterns from the wrinkles and vertical patterns from the visible lines). In this chapter, two main contributions are proposed in terms of feature extraction:

- A new feature extraction method called Multi-scale Sobel Angles Local Binary Pattern (MSALBP), and
- An enhanced method named Enhanced Local Line Binary Pattern (ELLBP).

Briefly, MSALBP firstly detects the Sobel vertical and horizontal edges from the FT. Then, it combines them according to their direction angles. A Multi-Scale Local Binary Pattern (MSLBP) is fused with the resulting image. The resulting appearances are formed into non-overlapping blocks and statistical calculations are implemented based on the Coefficient of Variance (COV) to form a texture vector. The texture vectors of the five fingers (thumb, index, middle, ring and little) are concatenated and used as inputs to an Artificial Neural Network (ANN). On the other hand, the ELLBP is an enhanced version of the Local Line Binary Pattern (LLBP). It is based on fusing the vertical and horizontal textures according to the weighted summation rule, which is useful to describe the FTs. Again, the resulting ELLBP images are blocked, then the COV is calculated for each block and the

feature vector is prepared for the ANN. A Probabilistic Neural Network (PNN) is applied to both approaches as a multi-classifier to perform the verification. Extensive experiments have been carried out to validate the proposed methods.
The organization of this chapter after the introduction is as follows: Section 2 explains the basic Local Binary Patterns (LBP) method. Section 3 illustrates some modified LBP methods. Section 4 describes the proposed feature extraction approaches. Their evaluations, results and discussions are given in Section 5. Finally, the summary of the overall chapter is declared in Section 6.

5.2 Basic Local Binary Pattern

The basic LBP was firstly introduced in [123] as a method of texture analysis. It has been used in various fields such as face recognition [41], face expression recognition [122] and object detection [163]. Fundamentally, the image is first divided into 3×3 sub-blocks. In each sub-block a comparison is carried out between the value of the center pixel and the values of its surrounding 8 neighbour pixels. The result of this comparison is a logical number; if the center value is smaller than the neighbour pixel value then a '0' is assigned to this location and if the center value is greater than or equal to the neighbour pixel value a '1' is put in this location. After that, a weighted sum equation is applied to convert the binary number to the decimal code. An example of the LBP code is shown in Fig. 5.1.
The following equation can be considered to calculate the LBP code [123]:

$$LBP = \sum_{p=0}^{7} s(g_p - g_c)2^p \tag{5.1}$$

where: g_c is the center pixel of the 3×3 sub-block, g_p is the p-th circular surrounding neighbour pixels and the LBP transformation function s is denoted as:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{5.2}$$

The resulting LBP image is usually divided into non-overlapped blocks. The histogram is calculated for each part and the histogram bins are concatenated to produce a single vector, which is considered as a feature vector of the original

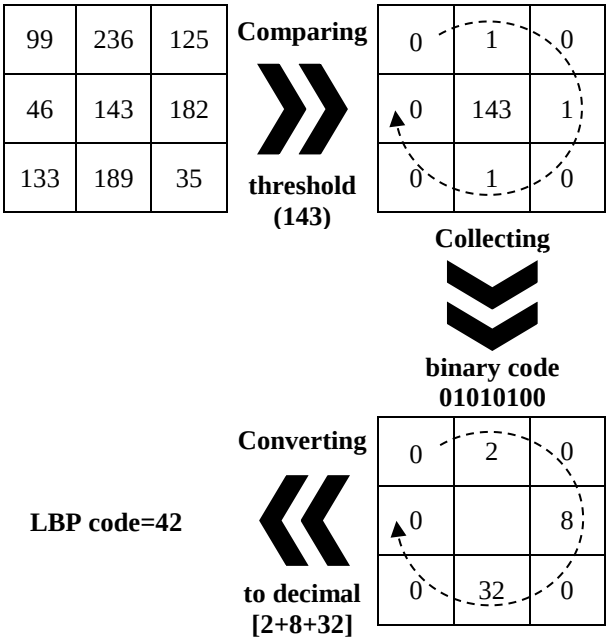


Figure 5.1: An example of computing the LBP code for a 3×3 window

image. However, some recent papers suggested to use the resultant image of the LBP directly instead of generating the histograms such as [25, 116, 126, 164]. Also, utilizing the histograms may result in losing important spatial information [116]. Moreover, the LBP feature vector based histogram is generally too large [42]. In this work the generated image codes have been used as the basis for texture patterns and a statistical calculation is employed to collect the features.

Although it has been confirmed that one of the best feature extraction methods which is illumination invariant is the LBP method, different enhancements are given for this effective method.

5.3 Enhanced Local Binary Patterns

5.3.1 Multi-Scale Local Binary Patterns

To modify the LBP, Ojala *et al.* in [124] proposed an MSLBP operator $MSLBP_{P,R}$ using a circular neighbourhood of pixels having different spatial sampling P and different radius R. The LBP values for the re-sampled pixels which are not located on the original grid are calculated by using the bilinear interpolation [124]. Fig. 5.2 shows different MSLBP operators.

Hence, the original LBP Equation (5.1) has been modified to the following MSLBP

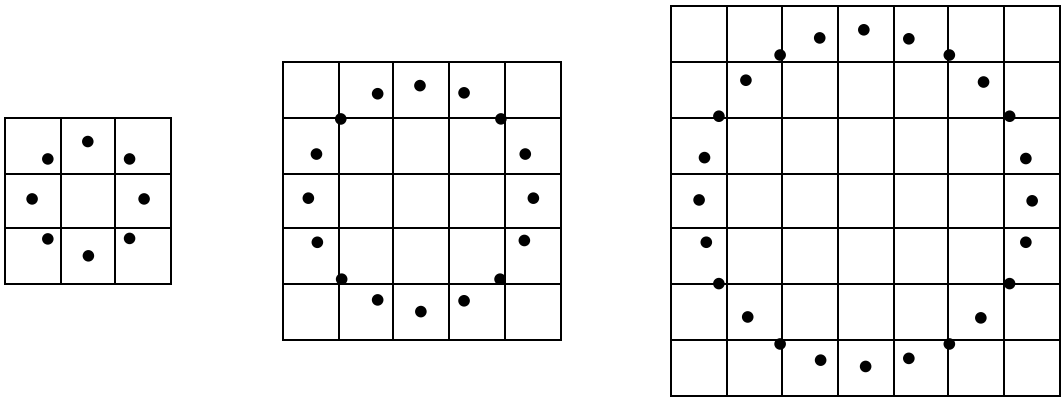


Figure 5.2: Different MSLBP operators, from the left $MSLBP_{8,1}$, $MSLBP_{16,2}$ and $MSLBP_{24,3}$

equation [124]:

$$MSLBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{5.3}$$

where g_p is the p-th gray level of the sampled value, g_c is the center value and the transformation function $s(x)$ is defined as in Equation (5.2).

5.3.2 Improved Local Binary Pattern

Jin *et al.* [11] suggested that the centre pixel in the basic LBP may provide important information. Therefore, an Improved Local Binary Pattern (ILBP) was described to include the centre pixels according to Equation (5.4) [11]:

$$ILBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - \bar{g})2^p + s(g_c - \bar{g})2^P \tag{5.4}$$

where: $\bar{g}$ is the average of the gray values for the pixels in the sub-block matrix and

$$s(x) = \begin{cases} 1 & , \quad x > 0 \\ 0 & , \quad x \leq 0 \end{cases} \tag{5.5}$$

As it can be seen from Equation (5.4) the centre pixel has the largest weight and

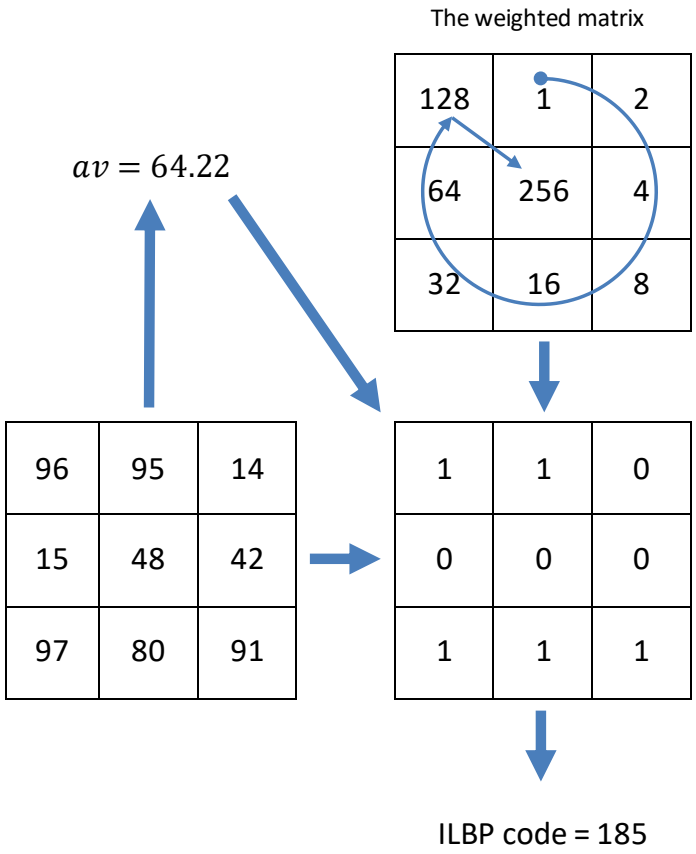


Figure 5.3: An example of computing the ILBP code for a 3×3 window according to [11]

the threshold comparison is performed with the average value. An example of the ILBP is given in Fig. 5.3.

5.3.3 Simplified Local Line Binary Pattern

Tao and Veldhuis [125] proposed a Simplified Local Binary Pattern (SLBP) method to overcome certain illumination problems. Simply, it is assumed that there should be no direction weights to calculate the LBP code. Thus, the image pixels will collect values between 0 to 8 as shown in Equation (5.6) [125]:

$$SLBP = \sum_{p=0}^{7} s(g_p - g_c) \tag{5.6}$$

where $s(x)$ is defined as in Equation (5.5).

One advantage of the SLBP is that the same SLBP code can be calculated from any direction as it is shown in Fig. 5.4. In otherwords, it is dimensionless [125].

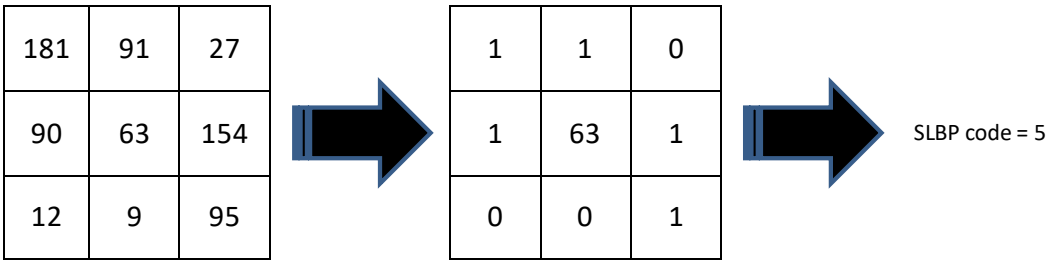


Figure 5.4: An example of the SLBP operator

However, the variances among the SLBP code values are too small (between 0 to 8) and this may cause insufficient collected information to describe the textures.

5.3.4 Three-Patch Local Binary Pattern and Four-Patch Local Binary Pattern

More advanced LBP methods have been suggested by Wolf *et al.* [12] called Three-Patch Local Binary Pattern (TPLBP) and Four-Patch Local Binary Pattern (FPLBP) respectively. Fig. 5.5 and Fig. 5.6 illustrate the TPLBP and FTLBP operators, respectively. Generally, the TPLBP in Fig. 5.5 is consists of a $w \times w$ patches, a centre patch with a centre pixel and organized $S = 8$ surrounded patches forming a circle shape around a radius r. A parameter α is chosen in order to determine the certain patches, which are specified to take apart in the calculations. A distance function is calculated between the values of any two surrounded patches and the values of the centre patch, from this description the name of the Three-Patch Local Binary Pattern is inspired. This operation will be considered in generating the TPLBP code [12]. On the other hand, the FPLBP in Fig. 5.6 is consists of two groups of patches each with a size of $w \times w$ are distributed in two circles around radii $r1$ and $r2$. Each circle has the same number of S patches and has the same pixel centre. A distance function is generated between the values of two symmetric patches from the outer circle and the values of two symmetric patches from the inner circle in a clockwise direction, from this concept the name of the Four-Patch Local Binary Pattern is used. This operation will be utilized in producing the FPLBP code [12].

The TPLBP and FPLBP codes can be calculated according to the following equations, respectively [12]:

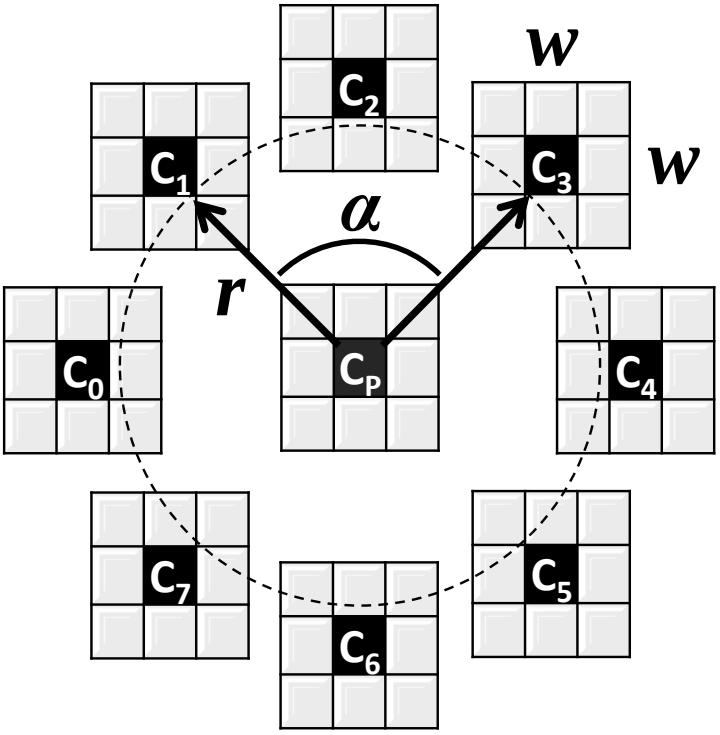


Figure 5.5: The main diagram of the TPLBP operator [12]

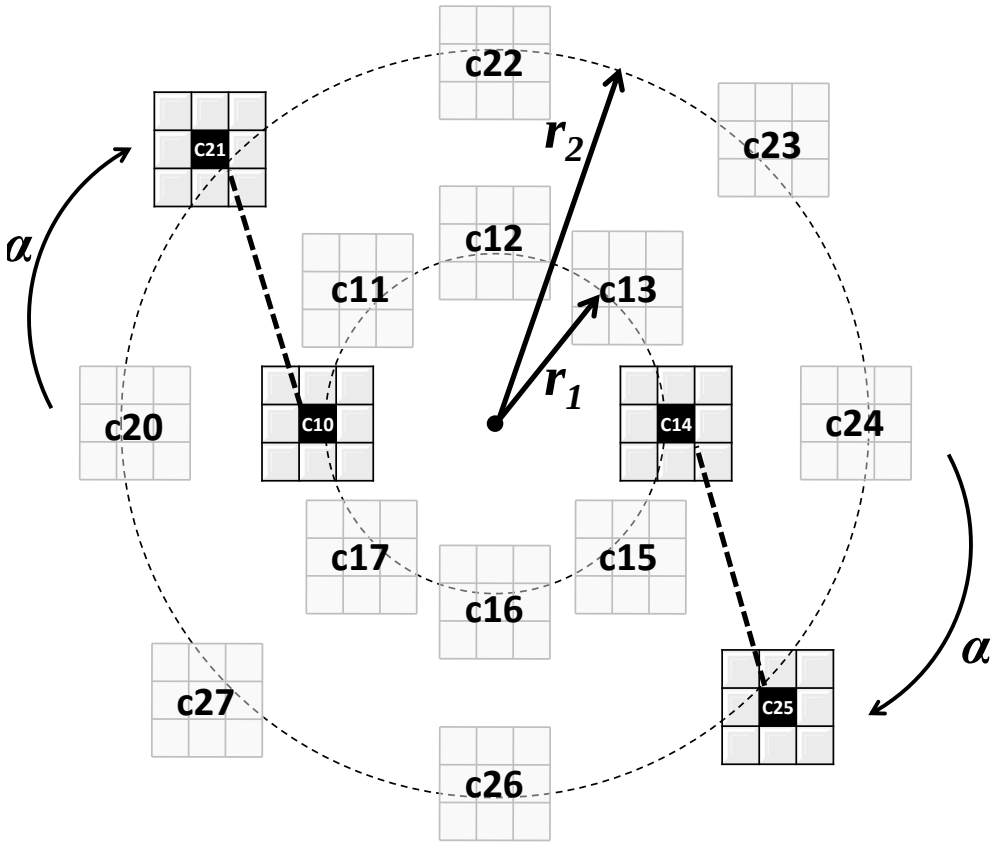


Figure 5.6: The main diagram of the FPLBP operator [12]

$$TPLBP_{r,S,w,\alpha}(p) = \sum_{i}^{S} f(d(C_i, C_p) - d(C_{i+\alpha \mod S}, C_p))2^i \tag{5.7}$$

$$\begin{aligned} FPLBP_{r_1,r_2,S,w,\alpha}(p) = \sum_{i}^{S/2} f(&d(C_{1i}, C_{2,i+\alpha \mod S}) \\ &-d(C_{1,i+S/2}, C_{2,i+S/2+\alpha \mod S}))2^i \end{aligned} \tag{5.8}$$

where: C_i and $C_{i+\alpha \mod S}$ are the two patches along the surrounding circle, C_p is the centre of the patch, $d(.,.)$ is the distance function between the two patches and f is denoted according to a specific threshold τ as:

$$f(x) = \begin{cases} 1 & \text{if} \quad x \geq \tau \\ 0 & \text{if} \quad x < \tau \end{cases} \tag{5.9}$$

Examples of the TPLBP and FPLBP are given below [12]:

$$\begin{aligned} TPLBP_{r,8,3,2}(p) = &f(d(C_0, C_p) - d(C_2, C_p))2^0 + f(d(C_1, C_p) - d(C_3, C_p))2^1 \\ &+f(d(C_2, C_p) - d(C_4, C_p))2^2 + f(d(C_3, C_p) - d(C_5, C_p))2^3 \\ &+f(d(C_4, C_p) - d(C_6, C_p))2^4 + f(d(C_5, C_p) - d(C_7, C_p))2^5 \\ &+f(d(C_6, C_p) - d(C_0, C_p))2^6 + f(d(C_7, C_p) - d(C_1, C_p))2^7 \end{aligned} \tag{5.10}$$

$$\begin{aligned} FPLBP_{r_1,r_2,8,3,1}(p) = &f(d(C_{10}, C_{21}) - d(C_{14}, C_{25}))2^0 \\ &+f(d(C_{11}, C_{22}) - d(C_{15}, C_{26}))2^1 \\ &+f(d(C_{12}, C_{23}) - d(C_{16}, C_{27}))2^2 \\ &+f(d(C_{13}, C_{24}) - d(C_{17}, C_{28}))2^3 \end{aligned} \tag{5.11}$$

As it has been demonstrated in Figs. 5.5 and 5.6, Wolf *et al.* [12] suggested dealing with a patch of sub-blocks instead of using the pixels of the sub-blocks to produce the texture image. The main drawback in such operators is that they will lead to losing the micro textures from the original images. Increasing the number of patches from three to four rather than increasing the radius between the patches could lead to losing more textures and wasting effective features.

5.3.5 Local Line Binary Pattern

A Local Line Binary Pattern (LLBP) which is suggested in [126] is an efficient method to analyse horizontal and vertical textures. This operator considers the patterns in both horizontal and vertical directions. See Fig. 5.7 where a matrix of pattern values is given; horizontal and vertical vectors are specified then analysed and a final LLBP code is produced from the results of the horizontal and vertical patterns calculations. The LLBP values can be calculated according to the following equations [126]:

$$LLBP_h(N,c) = \sum_{n=1}^{c-1} s(h_n - h_c)2^{(c-n-1)} + \sum_{n=c+1}^{N} s(h_n - h_c)2^{(n-c-1)} \tag{5.12}$$

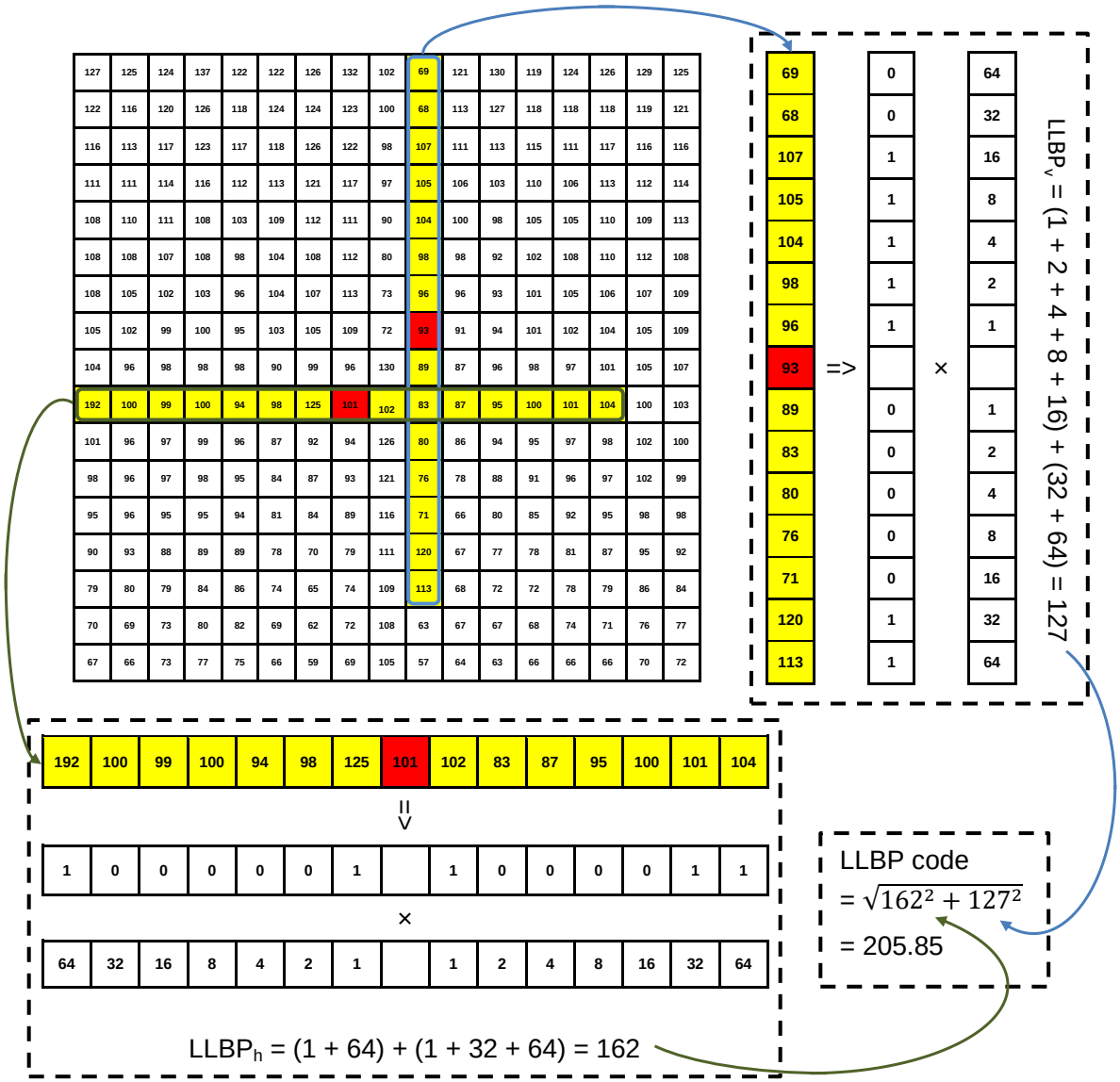


Figure 5.7: An example of the LLBP operator with the vector length of 15 pixels

$$LLBP_v(N,c) = \sum_{n=1}^{c-1} s(v_n - v_c)2^{(c-n-1)} + \sum_{n=c+1}^{N} s(v_n - v_c)2^{(n-c-1)} \tag{5.13}$$

$$LLBP_m = \sqrt{LLBP_h^2 + LLBP_v^2} \tag{5.14}$$

where: $LLBP_h$ represents the horizontal direction value, $LLBP_v$ represents the vertical direction value, $LLBP_m$ represents the magnitude value between the horizontal and vertical directions, N is the vector length in pixels, c is the position of the centre pixel = $\lceil \frac{N}{2} \rceil$ where $\lceil . \rceil$ denotes the ceiling operator, h_c represents the centre of the horizontal vector, v_c represents the centre of the vertical vector, h_n represents the pixels along the horizontal vector, v_n represents the pixels along the vertical vector and $s(x)$ represents the same function as in the LBP, namely Equation (5.2).

Moreover, Petpon and Srisuk [126] illustrated that the best results can be obtained for the vector lengths of N = 13, 15, 17 or 19; because they roughly cover the possible patterns to the maximum grayscale value (255).

5.3.6 Gradient Directional Pattern

Ahmed [127] has generated a robust method called the Gradient Directional Pattern (GDP). It is less sensitive to noise than the LBP. In this descriptor the Sobel angle orientations are considered in degrees. It utilizes a comparison tolerance function with the center pixel in the LBP operator. See Fig. 5.8, where an examples of a 3×3 matrix is shown, a comparison is implemented between the value of the centre pixel $\pm$ a tolerance threshold (22.5), a binary code is established from the previous comparisons and a GDP code is computed as a decimal value from the binary code. The following equations are considered in the GDP operator [127]:

$$GDP = \sum_{p=0}^{P-1} s(GD_p, GD_c)2^p \tag{5.15}$$

$$s(GD_p, GD_c) = \begin{cases} 1, & GD_c - \beta \leqslant GD_P \leqslant GD_c + \beta \\ 0, & \text{otherwise} \end{cases} \tag{5.16}$$

where: GD_p represents the direction of the surrounding pixels in the sub-block kernel, GD_c represents the centre pixel value of the sub-block kernel and β is the determined threshold.

The essential problem in this work is that the comparison function based on the tolerance in Equation (5.16) causes loss of texture from the image. In particular, if the comparison value is more or less than the center pixel ± the tolerance, this

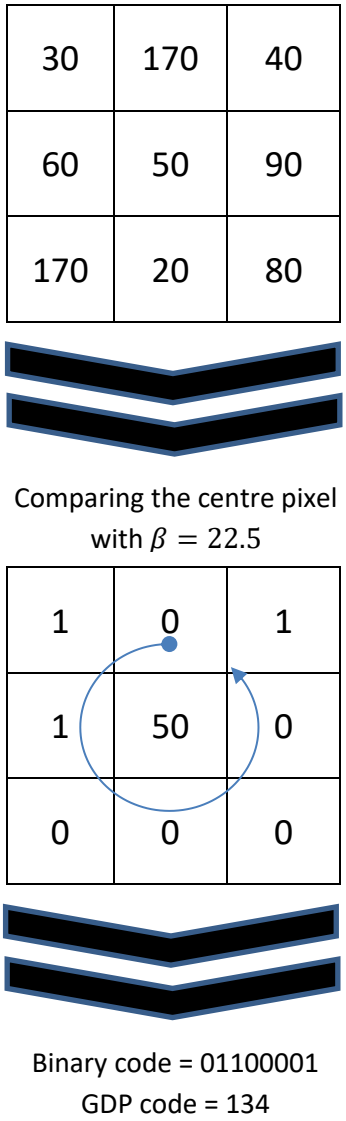


Figure 5.8: An example of computing a GDP code

value would be set to zero, where it might have a texture. Moreover, the value of β was denoted as a user-specified tolerance and it was found empirically in [127]. In this work, the GDP operator has been simulated and used in the comparisons. The best β values for the three employed databases have been determined here after many experiments.

5.3.7 Improved Local Binary Pattern Neighbours

Liu *et al.* [116] have suggested an Improved Local Binary Pattern Neighbours (ILBPN) method to analyse inner knuckle prints. The ILBPN is followed by calculating uniform LBP values for each pixel. Then, binary images have been generated for each LBP uniform value. So, a massive number of binary images have been generated, where for each sample 59 binary images were stored in the template and the same number of binary images was produced for each sample during the testing stage. These binary images were used as the basis of the matching and recognition decision. In this study, the important part of this descriptor for the comparison is the ILBPN approach.

The ILBPN operator has been described as a horizontal window of 9 pixels. The pixel value in the middle will be recalculated based on the LBP method. A simple

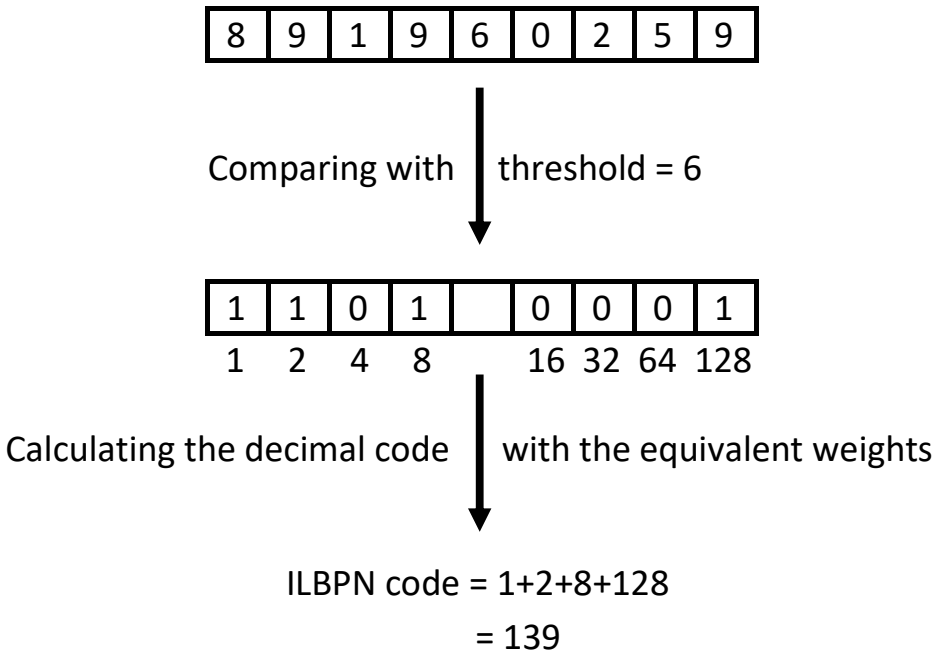


Figure 5.9: An example of the ILBPN operator

example of the ILBPN operator is demonstrated in Fig. 5.9. In this example a horizontal vector of pattern values of pixels is given. It consists of 9 pixels. So, the centre pixel is located in position 5. The value of this pixel is used as a threshold. Any pixel with a greater or equal value to this threshold is set to '1', otherwise it equals to '0'. Converting from binary to decimal is computed according to specific weights. Finally, the decimal code is used as an ILBPN code.
This operator is more likely to be used for the patterns in a vertical direction as illustrated in [116].

5.3.8 Local Gradient Coding and Local Gradient Coding-Horizontal Diagonal

The authors in [13] have proposed a Local Gradient Coding (LGC) operator based on the gradient difference around the centre pixel. It has been designed by computing the gradient between the horizontal vectors, the gradient between the vertical vectors and the gradient between the diagonal as described in Equation (5.17):

$$\begin{aligned} LGC_{P,R} = s(g_1 - g_3)2^7 + s(g_4 - g_5)2^6 + s(g_6 - g_8)2^5 + \\ s(g_1 - g_6)2^4 + s(g_2 - g_7)2^3 + s(g_3 - g_8)2^2 + \\ s(g_1 - g_8)2^1 + s(g_3 - g_6)2^0 \end{aligned} \tag{5.17}$$

This operator uses the transformation condition function as in Equation (5.5).
Moreover, the authors have found that reducing the features by neglecting the vertical gradients does not affect the facial expression recognitions of their employed database. So, in the same work they have suggested the Local Gradient Coding-Horizontal Diagonal (LGC-HD), where the following equation has been considered for just the horizontal and diagonal gradients [13]:

$$\begin{aligned} LGC_{P,R} = s(g_1 - g_3)2^4 + s(g_4 - g_5)2^3 + s(g_6 - g_8)2^2 + \\ s(g_1 - g_8)2^1 + s(g_3 - g_6)2^0 \end{aligned} \tag{5.18}$$

The gradient pixel positions are illustrated in Fig. 5.10. A simple mathematical example of the LGC is demonstrated in Fig. 5.11 based on the gradient pixel positions, where all the possibilities of the gradients (the vertical, horizontal and

diagonal) around the centre pixel are considered.

g_1	g_2	g_3
g_4	g_c	g_5
g_6	g_7	g_8

Figure 5.10: The gradient pixel positions in the LGC kernel [13]

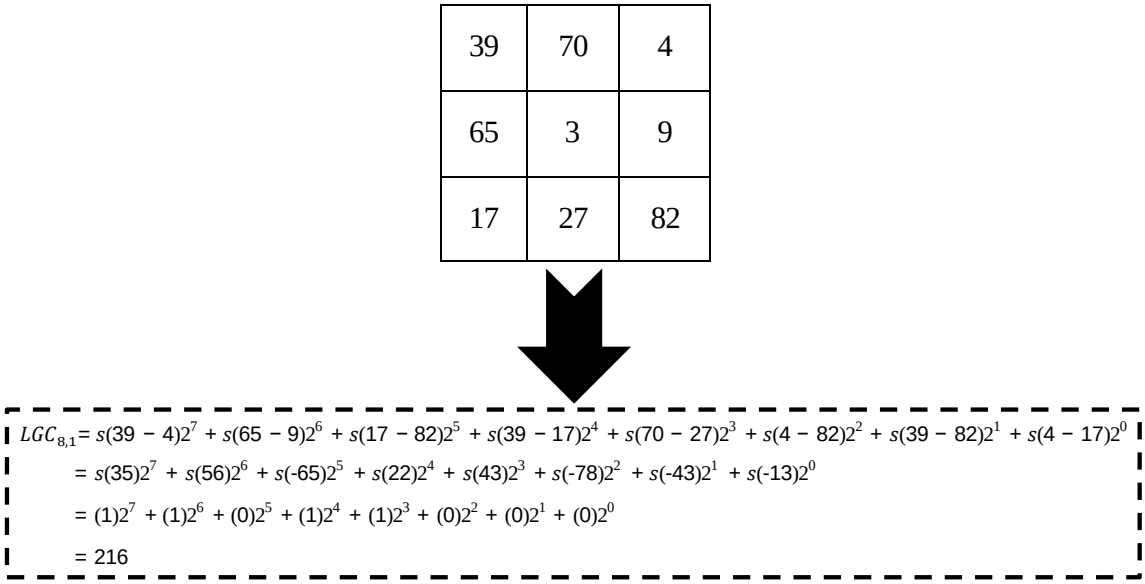


Figure 5.11: A simple mathematical example of calculating the LGC code

In this study, both operators (the LGC and the LGC-HD) have been examined and applied in the comparison.

5.4 Proposed Feature Extraction Approaches

5.4.1 Multi-scale Sobel Angles Local Binary Pattern

In this approach, the features are extracted from each finger using a new proposed descriptor. This descriptor is constructed based on multiple processing stages as shown in Fig. 5.12.

The proposed feature descriptor which is called MSALBP is designed to produce an effective feature extraction for the FTs. It starts by filtering the finger images

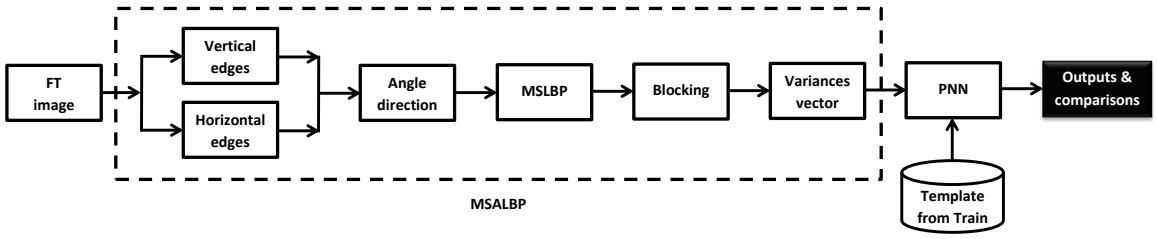


Figure 5.12: The block diagram of the suggested scheme for the MSALBP feature extraction and the verification operation

with the Sobel operator masks, where each operator mask is related to a direction (vertical or horizontal). Both direction operators are shown in Fig. 5.13a and Fig. 5.13b, respectively [165].

+1	+2	+1
0	0	0
-1	-2	-1

(a)

-1	0	+1
-2	0	+2
-1	0	+1

(b)

Figure 5.13: The Sobel operator masks: (a) The horizontal edge detection operator and (b) The vertical edge detection operator

The sign directions of the Sobel operators are empirically determined to obtain consistent performance and make fair comparisons. To combine the horizontal and vertical features of an image, let Gx be the convolved image with the vertical edge detection mask and Gy be the convolved image with the horizontal edge detection mask. Both filtered images can be fused together by using one of the following equations [165]:

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{5.19}$$

$$\theta = atan2(Gy, Gx) \tag{5.20}$$

where $|G|$ represents the amplitude of the gradient, whereas θ represents the angle

direction of the edges and $atan2$ represents the four-quadrant inverse tangent described in [166] but used in Matlab for the range of $[-\pi, \pi]$.

It has been cited that the amplitude/magnitude equation is not robust to provide directional information as in a phase/angle calculation [167, 168], whereas angle features can efficiently describe the gradients of certain patterns with less sensitivity to the pixel level values than the amplitude features. In this work, the angle direction matrix has been chosen to generate the new MSALBP operator. It attained better results than the amplitude as will be demonstrated in the results and discussion section.

Fig. 5.14 shows examples of the five FT images with their horizontal edges, vertical edges and the $\text{MSLBP}_{8,2}$ images of the angle images.

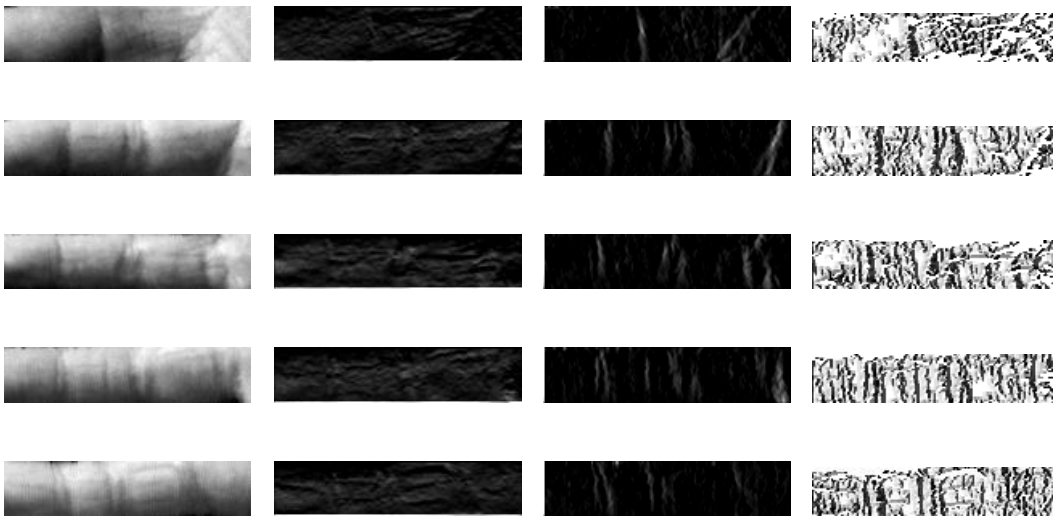


Figure 5.14: Image analysis of the MSALBP operator: Each row represents a ROI of a finger; from the top: thumb, index, middle, ring and little finger respectively. While, the first column is assigned for the original FT images, the second column shows the horizontal edge images, the third shows the vertical edge images and the last row represents the $\text{MSLBP}_{8,2}$ of the angle images

Henceforth, the following equation can be used to fuse the operator with the edge angles direction image, resulting in a modified version of the MSLBP:

$$MSALBP_{P,R} = \sum_{p=0}^{P-1} s(gt_p - gt_c)2^p \tag{5.21}$$

where $s(x)$ is as in equation 5.2; gt_p and gt_c represent the p-th neighbour and the center pixels of each sub-block after the Sobel angle direction image, respectively.

In order to obtain the best implementation for the FT patterns, different values of radius and neighbour parameters have been examined.

Each input image is partitioned into non-overlapped matrices with a fixed size of 5×5. The COV value is calculated for each block according to Equations (5.22), (5.23) and (5.24), respectively [169]:

$$M_{bl} = \frac{1}{n} \sum_{i=1}^{n} bl_i \tag{5.22}$$

$$STD_{bl} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (bl_i - M_{bl})^2} \tag{5.23}$$

$$COV_{bl} = \frac{STD_{bl}}{M_{bl}} \tag{5.24}$$

where: n is the number of pixels in each block, bl is a block of 5×5 pixels, i is the pixels' pointer in each block, M_{bl} is the average value of the block pixels, STD_{bl} is the standard deviation value of the block pixels and COV_{bl} is the coefficient of variance of the block pixels.

The COV value has the following advantages: it reduces the image size, where each image block has been reduced to one value; it is easy to calculate; it efficiently describes the variances between the pixel values [26]; all values are small and positive; the variances between the features for the same subject are well described and the variances between the features for different subjects are well described [27].

All COV values of all fingers are arranged into a 1D vector. This vector is then applied as the input to the PNN to classify the variances between the data.

It is true that this is not the first time the Sobel method is used with the LBP as in [127,170,171]. However, the structure is different, where in [170,171] the Sobel edge images were used with different operator directions and then each of their vectors were concatenated together to construct the feature vector. This resulted in very large vectors, where in [170] the feature vector size was equal to 2124 values and in [171] the feature vector size consisted of 4248 values. The GDP in [127] was covered in Section 5.3.6 and its operator is applied in this work for the comparison.

5.4.2 Enhanced Local Line Binary Pattern

Due to the fact that the textures of the inner finger surfaces mainly consist of vertical and horizontal patterns, the LLBP operator is developed as an efficient method to describe these features termed the ELLBP.

The main problem in the LLBP can be found in its amplitude Equation (5.14). As mentioned in Subsection 5.4.1, the amplitude computations are not appropriate to provide directional information as in the gradient (or phase) calculations. They can be influenced by noise, brightness and range value according to [172], therefore, it cannot give effective description of image textures. On the other hand, fusing the vertical and horizontal vectors according to the weighted summation rule is efficient to describe the FTs. Thus, Equation (5.25) is suggested to be used instead of Equation (5.14):

$$ELLBP_m = ((v_1 \times LLBP_h) + (v_2 \times LLBP_v))/2 \tag{5.25}$$

where v_1 and v_2 are the directional strengths ($v_1 + v_2 = 1$); these two parameters can be used to control the density of the two directions. The reason for dividing by 2 is to get the average value between the weighted summation of the values calculated from the vertical and horizontal lines; this covers the possible patterns from the minimum grayscale value (0) to the maximum grayscale value (255). However, this can be neglected because of using the COV calculations, but in the ideal operator it is better to be used.

An example is depicted in Fig. 5.15, where FT data acquired from a small region of a single finger is analysed according to the ELLBP method. In this figure; true pixel values of the small FT region are enlarged; a middle pixel is selected to be analysed and a size of 17 pixels is determined from both horizontal and vertical directions; the vertical and horizontal vectors are acquired then analysed separately considering the same central pixel and finally, the ELLBP code can be calculated by using the suggested new equation.

As three databases have been employed in the later evaluation and each one of them consists of different characteristics, so, it is not easy to establish the v_1 and v_2 values. For instance, the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9] is a low resolution database acquired by using electromagnetic

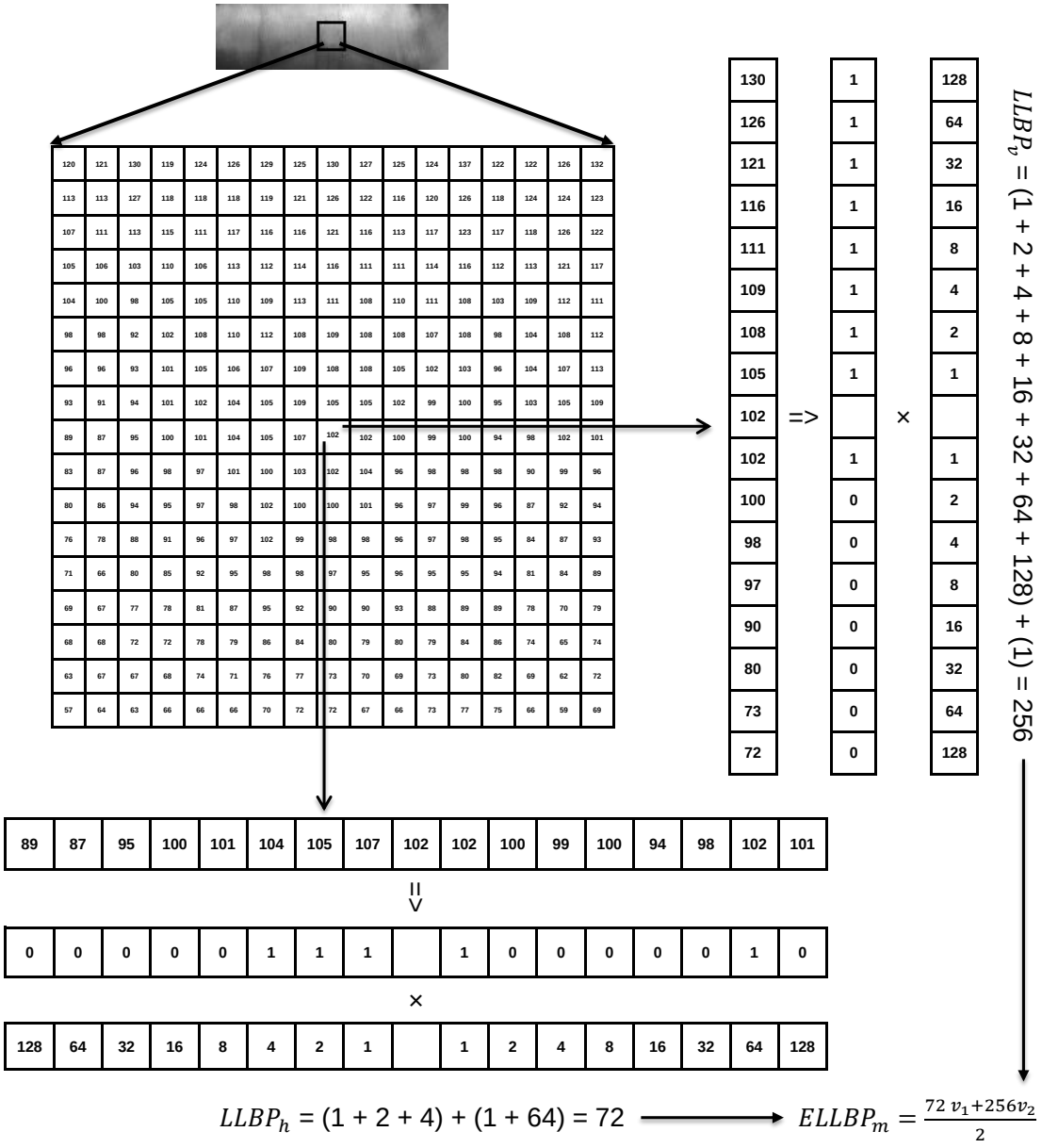


Figure 5.15: A demonstrated example of the suggested ELLBP (N=17)

spectrum sensors, while The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67] is a very low resolution database acquired for hands located at a long distance from the capturing device. It has been investigated that for choosing the values of v_1 and v_2 the following useful idea can be utilized:

Dividing the training patterns into training and validation parts. Then, the weighted summation will be examined in the training phase according to the partitioned parts. After that, the best values can be used during the testing phase. This idea has been inspired from [173], where three novelties to acquire the weighted summation values have been introduced. All of these methods were based on using only the training

data. However, massive calculations have been applied in [173] to collect the precise values of the weights.

Samples of the ELLBPs for five finger images are given in Fig. 5.16. Similar COV computations which have been described in the MSALBP (Subsection 5.4.1) are applied to the ELLBP image after dividing it into non-overlapped blocks of equal sizes. The computed COV values are organized into a 1D vector for the PNN step.

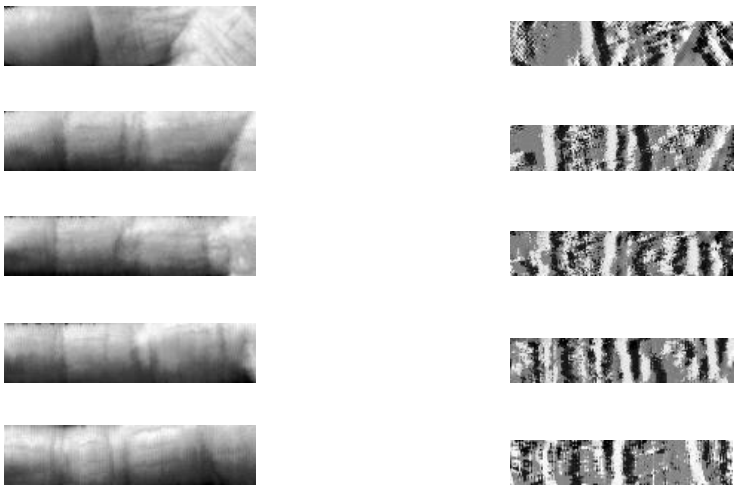


Figure 5.16: Example of five ROI finger images and their ELLBPs: the first column in the left represents ROI of each finger and the last column represents the ELLBP image for each finger. While the first row is for the thumb, the second row is for the index finger, the third row is for the middle finger, the fourth row is for the ring finger and the fifth row is for the little finger

5.5 Results and Comparisons

5.5.1 Evaluating the Suggested MSALBP Feature Extraction

This subsection investigates different aspects in the proposed MSALBP with the traditional PNN. After the Sobel direction filters two output components can be computed: the amplitude and angle. The angle has been selected as it gives the best results in terms of the Equal Error Rate (EER) as shown in Table 5.1.

This table clearly indicates that using the Sobel angle directions in the MSALBP gives better results than using the amplitude. This is because the amplitude is affected by the illumination, imaging contrast and random noise from the

Table 5.1: Sobel amplitude values versus Sobel angle directions

PolyU3D2D database		
Sobel combination method	**LBP parameters**	**EER**
MSALBP (amplitudes)	P=8,R=1	2.15%
MSALBP (angles)	P=8,R=1	0.79%
IIT Delhi database		
Sobel combination method	**LBP parameters**	**EER**
MSALBP (amplitudes)	P=8,R=1	10.14%
MSALBP (angles)	P=8,R=1	2.03%
CASIAMS (S460) database		
Sobel combination method	**LBP parameters**	**EER**
MSALBP (amplitudes)	P=8,R=1	8%
MSALBP (angles)	P=8,R=1	5%

acquisition device while the angle directions is not. So, the angle patterns have more effective information than the amplitude patterns.

Secondly, the MSALBP is tested under different block sizes: 3×3, 5×5, 7×7, 9×9, 11×11, 13×13 and 15×15. The best block size is found to be 5×5 as it achieves the best EERs as given in Figs. 5.17 and 5.18 and varying the blocking size to more than or less than the suitable size will increase the verification error rate.

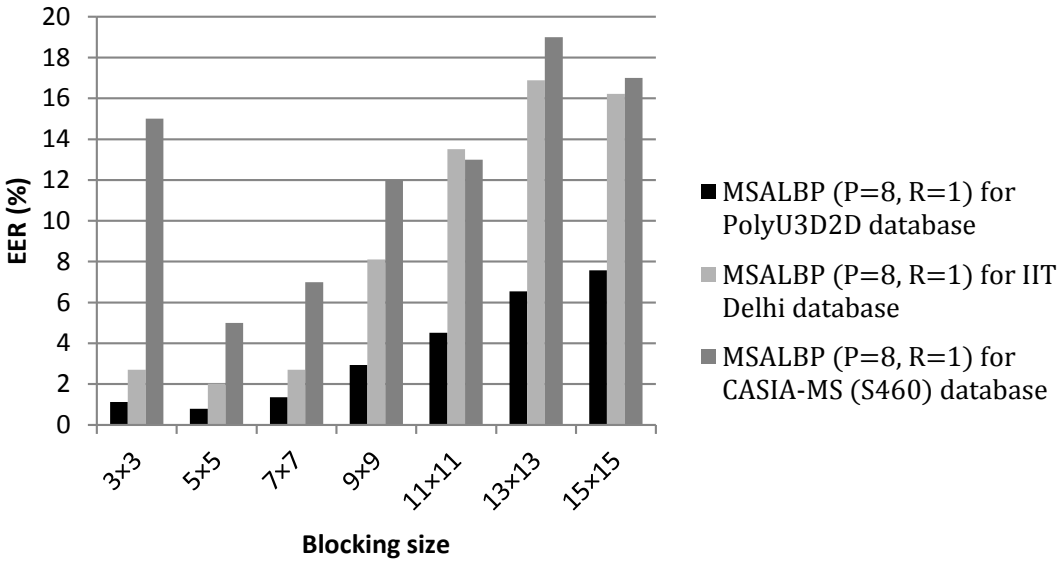


Figure 5.17: The performance of the $MSALBP_{8,1}$ by using various blocking sizes

The next evaluation aims to find the best parameters for the proposed MSALBP. Following [174], the testing parameters are: (P = 8 , R = 1), (P = 8 , R = 2), (P = 16 , R = 2), (P = 16 , R = 3) and (P = 24 , R = 3) respectively.

Table 5.2: The verification performance of the different MSALBP parameters

PolyU3D2D database		
Reference	**Parameters**	**EER**
Suggested approach	P=8 , R=1	0.79%
	P=8 , R=2	**0.68%**
	P=16 , R=2	0.79%
	P=16 , R=3	1.02%
	P=24 , R=3	1.58%
IIT Delhi database		
Reference	**Parameters**	**EER**
Suggested approach	P=8 , R=1	2.03%
	P=8 , R=2	2.03%
	P=16 , R=2	**1.35%**
	P=16 , R=3	4.05%
	P=24 , R=3	2.70%
CASIAMS database (S460)		
Reference	**Parameters**	**EER**
Suggested approach	P=8 , R=1	5%
	P=8 , R=2	**2%**
	P=16 , R=2	4%
	P=16 , R=3	4%
	P=24 , R=3	9%

It can be observed from Table 5.2 that using the MSALBP with a multi-scale of P = 8 and R = 2 parameters yielded the best verification performance for the PolyU3D2D and Spectrum 460nm (S460) from the CASIAMS databases compared with the other parameters such as P = 16, R = 2 and P = 16, R = 3. Whereas, applying the MSALBP with a multi-scale of P = 16 and R = 2 parameters attained the best verification performance for the Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66]. The fluctuating results in the EER values of the IIT Delhi database are due to the faulty acquired images as explained in Chapter 3. The general rule here is that increasing the radius R to more than 2 will miss micro-textures, whereas, decreasing the radius to 1 will include the micro-textures and the embedded noise as well. On the other hand, the number of surrounded pixels P is relating to the amount of processed information and this is the reason that the high resolution images in the IIT Delhi database required more surrounded neighbourhoods than the low resolution images in the S460 of CASIAMS and PolyU3D2D databases. It has been observed previously that reducing the number of the surrounding neighbourhood pixels and raising the value of the radius have the drawback of missing textures to some degree and whilst having a benefit of reducing the analysed data [174].

The best blocking size has been examined again after determining the best MSALBP parameters. Fig. 5.18 shows the performance of the different blocking size after using the MSALBP with its best parameters for each database.

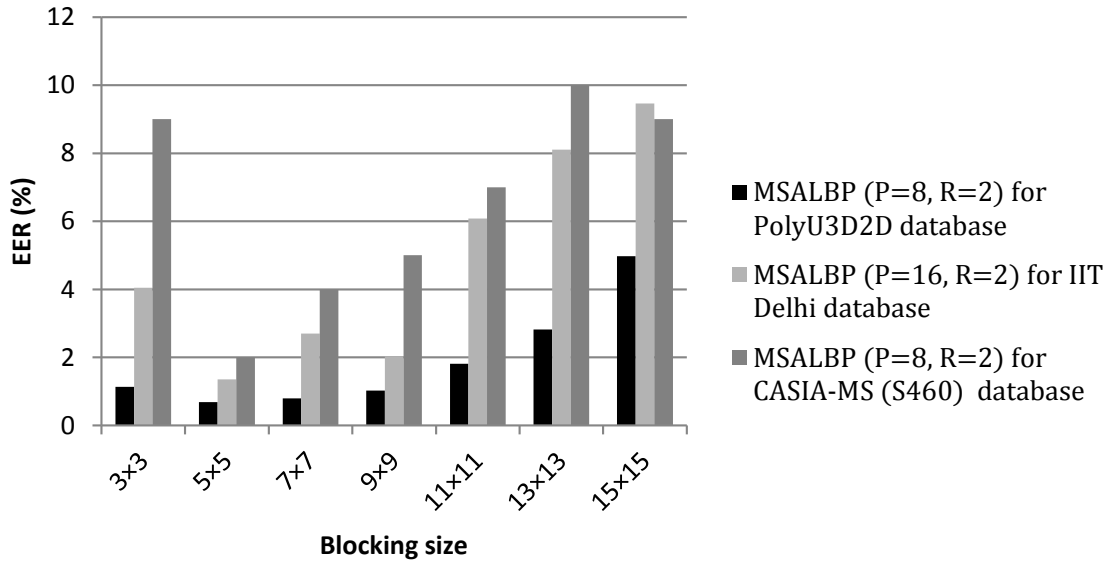


Figure 5.18: The performance of the best MSALBPs by using various blocking size, where the $MSALBP_{8,2}$ has attained the best features for the PolyU3D2D and S460 from the CASIAMS databases and $MSALBP_{16,2}$ has the best features for the IIT Delhi database

It is obvious from Fig. 5.18 that the best blocking size is still 5×5 as it has attained the best performance in all three databases. So, this blocking size has been confirmed to be the preferable choice.

5.5.2 Evaluating the ELLBP

The first essential part to be addressed in the ELLBP is the values of $v1$ and $v2$ in the weighted summation Equation (5.25). As mentioned, to solve this issue the training vectors were divided into training and validation patterns. Then, the weighted summation values were examined according to the partitioned vectors. This method has been executed in the three databases, where for each person four training patterns have been utilized for the training and one pattern has been applied for the validation, to calculate the best values of $v1$ and $v2$. This is shown in Figs. 5.19, 5.20 and 5.21.

Because each database has different specifications this caused variation in the weighted summation values of $v1$ and $v2$. For instance, the CASIAMS database is

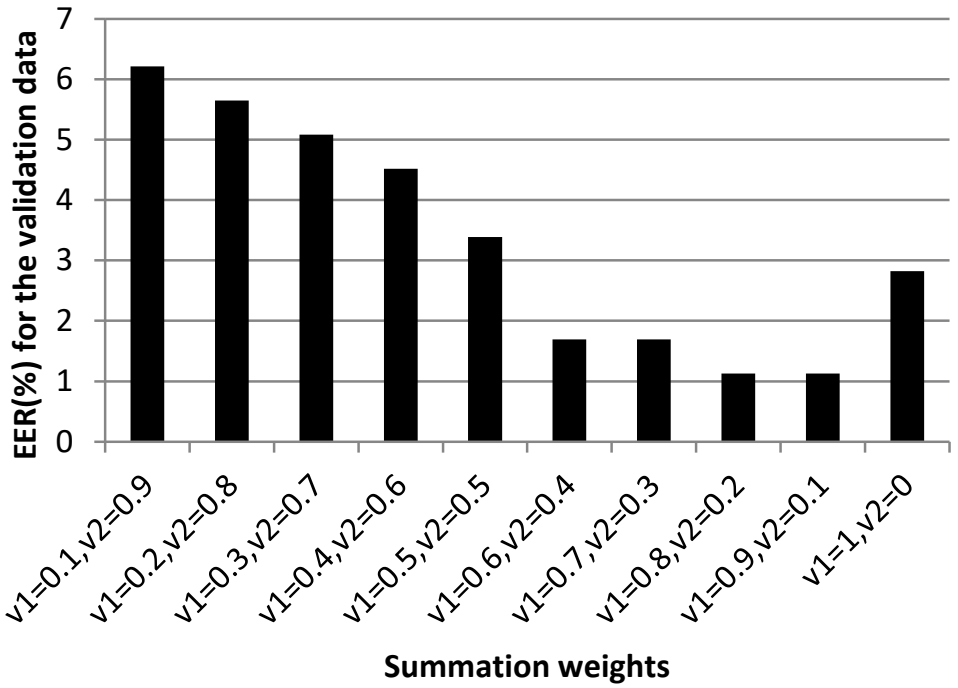


Figure 5.19: The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the PolyU3D2D database

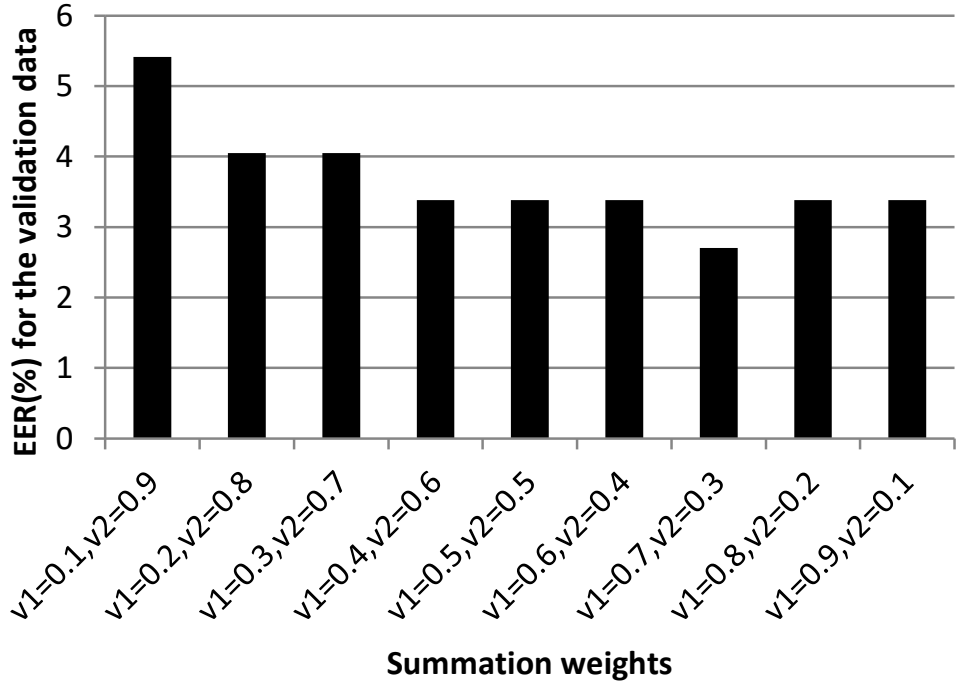


Figure 5.20: The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the IIT Delhi database

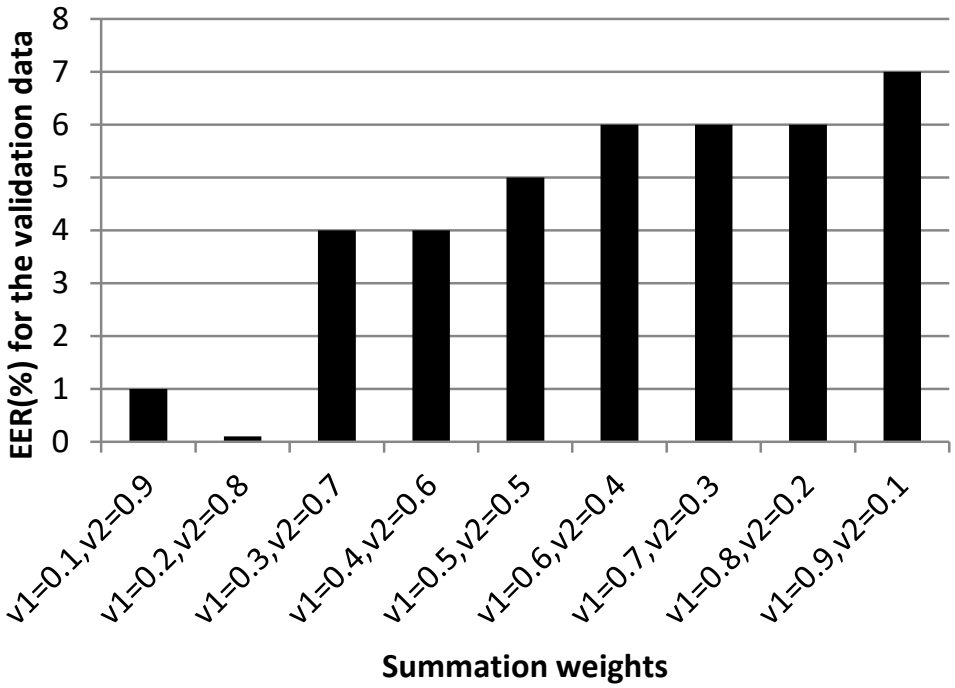


Figure 5.21: The performance of the different vertical and horizontal weights ($v1$ and $v2$, respectively) in the ELLBP(N=17) for the CASIAMS (S460) database

low resolution data and it is basically acquired under different spectral light sensors.

It can be seen from Figs. 5.19, 5.20 and 5.21 that the optimum values of $v1$ and $v2$ have different values between the three databases according to their specifications. Tuning these parameters by decreasing or increasing from the optimum value would influence the EER percentage to go far from its best performance. In both PolyU3D2D and IIT Delhi databases the vertical textures required increasing $v1$ and reducing $v2$, this is due to the clarity and effectiveness of the vertical knuckles versus the horizontal textures. On the other hand, increasing the horizontal texture by increasing $v2$ and reducing $v1$ effects the EERs in CASIAMS (S460) database; because this caused the horizontal features to dominates the vertical features. Sometimes, more than one $v1$ and $v2$ values have reported best EERs as in the PolyU3D2D databases. In this case any of these values can be used. Performances which have been recorded in [26] are considered in this work, that is, the performances of the weight values close to $v1 = 0.8$ and $v2 = 0.2$.

In the case of blocking size, similar results to the MSALBP have been obtained in the ELLBP (N=17). That is, the best blocking size of the ELLBP (N=17) can be

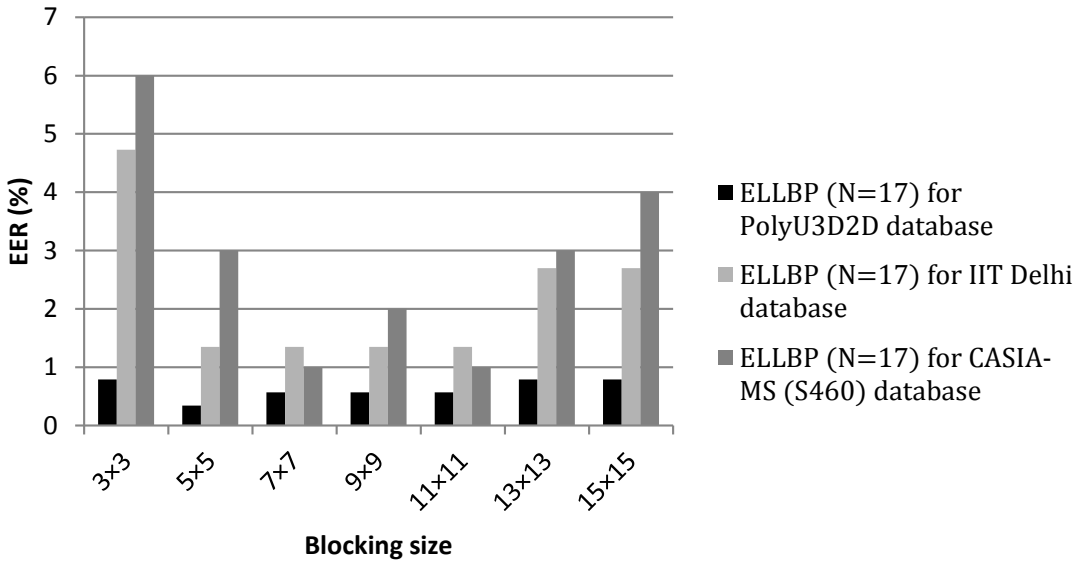


Figure 5.22: The performance of the ELLBP (N=17) by using various blocking sizes

considered to be equal 5×5 as this is obvious in Fig. 5.22 for the PolyU3D2D and IIT Delhi databases. However, unexpectedly the CASIAMS (S460) database has obtained unstable blocking results and this is due to the specifications of the FT in this database, where a blue light has been used in acquiring the hand images. Therefore, the FTs acquired by a spectral colour light expose different features from the FT specifications acquired by a normal light, as mentioned in Chapter 3. This leads to the next investigation of examining the different vector sizes of the ELLBP.

All the employed databases have been examined for various length of the ELLBP. Table 5.3 has been established to evaluate the effect of the ELLBP lengths on the FTs.

From this table, the ELLBP with the length N=17 has the best performance in the PolyU3D2D database. There is also another possibility with N=13, however, this length is not suitable in the IIT Delhi database. On the other hand, various changes in the ELLBP length for the IIT Delhi database almost achieve the lowest EER value. So, in general the FT images acquired under normal lightings can obtain the best EER when the length of the ELLBP is equal to 17 pixels. Furthermore, this length can cover the all possibilities of the grayscale values (0-255). Moreover, the very low resolution images would be influenced by tuning the ELLBP length as the small textures in such images are not clear enough to be considered. Whereas, the high resolution images might achieve the least EER even

Table 5.3: The EER results for ELLBP lengths of the all databases by using the PNN classifier

ELLBP length	**EER of PolyU3D2D Database**	**EER of IIT Delhi Database**	**EER of CASIAMS (S460) Database**
N=19	0.68%	2.03%	2%
N=17	0.34%	1.35%	3%
N=15	0.45%	1.35%	2%
N=13	0.34%	2.03%	2%
N=11	0.68%	1.35%	0%
N=9	0.79%	1.35%	0%
N=7	0.79%	1.35%	0%
N=5	1.24%	2.03%	3%
N=3	10.28%	8.11%	17%

after adjusting the length, because different size of textures are able to be analysed in such images. Again, the ELLBP with the length of N=17 seems to be the best choice for the acquired images by using the normal light.

Also, it can be seen from Table 5.3 that the S460 images from the CASIAMS database are clearly affected by the tuning length of the ELLBP. The reason for this is that the sensor light of the wavelength 460nm has enhanced the small horizontal textures of the FTs and this caused the small length of the ELLBP vector to be better. Surprisingly, the best EER here has been benchmarked to 0%. Nevertheless, choosing the best length for this type of the database is confusing, where the lengths: N=7, 9 and 11 pixels are compared in terms of the minimum EER. To overcome this issue a novel classifier termed Finger Contribution Fusion Neural Network (FCFNN) has been assigned to carry out the decision of the best ELLBP length of CASIAMS (S460) images. The FCFNN approach will be explained in Chapter 6 and the reason for using this method is that it can efficiently describe the contribution of the fingers in relation to the EERs. The relationships between the ELLBP lengths and their EERs by using the FCFNN technique is given in Table 5.4.

A nice and clear gradient can be observed in this table, where the ELLBP with the length of 7 pixels is the best EER at 0%. While, all other lengths have higher EER percentages. From this point, the blocking chart can be constructed again to obtain the best blocking size as shown in Fig. 5.23.

Obviously, Fig. 5.23 has again confirmed that the best blocking size to be assigned

Table 5.4: The EER results for ELLBP lengths of the CASIAMS (S460) Database by using the FCFNN classifier

ELLBP length	**EER of CASIAMS (S460) Database**
N=19	12%
N=17	6%
N=15	3%
N=13	3%
N=11	1%
N=9	1%
N=7	0%
N=5	4%
N=3	34%

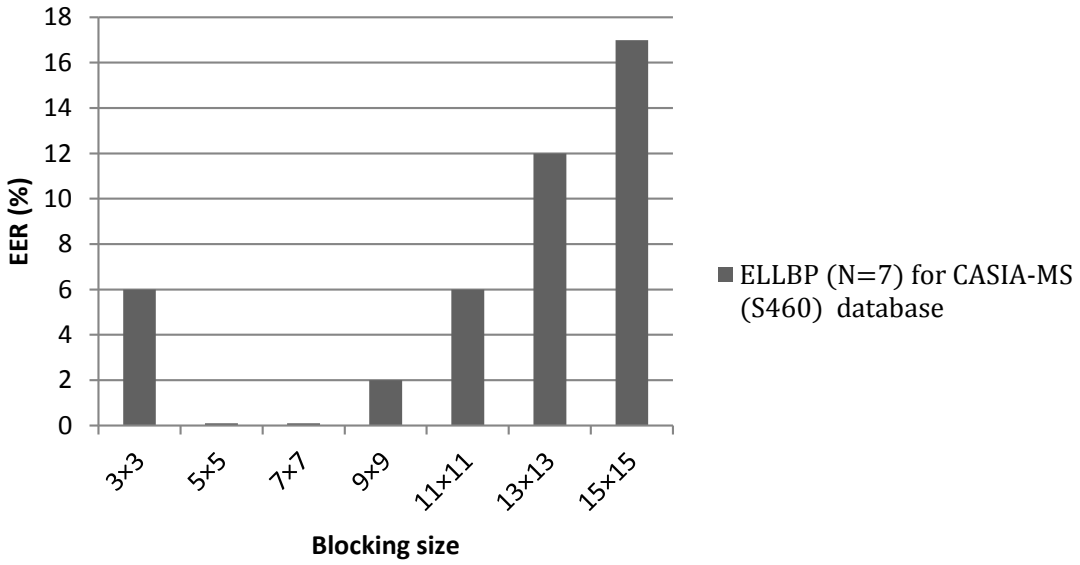


Figure 5.23: The performance of the ELLBP (N=7) for the CASIAMS (S460) by using various blocking size

is 5×5 in addition to the blocking size of 7×7, but this one cannot be considered as the best choice in the other databases. Furthermore, the best blocking size for the MSALBP was found consistently to be 5×5. Therefore, the blocking size of 5×5 pixels is the best choice as it has achieved the best EER performance in all three databases and in both feature extraction approaches.

5.5.3 Texture and Timing Comparisons with Related Work

In this study, comparisons between different LBP types have been established for the three databases, where each type has been simulated then implemented. For

Table 5.5: Comparisons between the verification performance of different LBP types for the FTs of the three databases

PolyU3D2D database			
Reference	**Method**	**Parameters**	**EER**
Jin *et al.* [11]	ILBP	P=8 , R=1	0.79%
Tao and Veldhuis [125]	SLBP	P=8 , R=1	1.47%
Wolf *et al.* [12]	TPLBP	w=3, r=2, S=8, α=5 and τ=0.01	1.47%
	FPLBP	w=3, r1=4, r2=5, S=8, α=1 and τ=0.01	9.38%
Petpon and Srisuk [126]	LLBP	N=19	1.24%
	LLBP	N=17	0.68%
	LLBP	N=15	0.79%
	LLBP	N=13	0.68%
Ahmed [127]	GDP	τ=4	1.36%
Liu *et al.* [116]	ILBPN	N=9	10.17%
Tong *et al.* [13]	LGC	n/a	1.24%
	LGC-HD	n/a	1.36%
Proposed approaches	MSALBP	P=8 , R=2	0.68%
	ELLBP	**N=17,v1=0.8,v2=0.2**	**0.34%**
IIT Delhi database			
Reference	**Method**	**Parameters**	**EER**
Jin *et al.* [11]	ILBP	P=8 , R=1	2.70%
Tao and Veldhuis [125]	SLBP	P=8 , R=1	2.70%
Wolf *et al.* [12]	TPLBP	w=3, r=2, S=8, α=5 and τ=0.01	6.76%
	FPLBP	w=3, r1=4, r2=5, S=8, α=1 and τ=0.01	15.54%
Petpon and Srisuk [126]	LLBP	N=19	4.05%
	LLBP	N=17	2.70%
	LLBP	N=15	2.70%
	LLBP	N=13	2.03%
Ahmed [127]	GDP	τ=4	2.70%
Liu *et al.* [116]	ILBPN	N=9	29.05%
Tong *et al.* [13]	LGC	n/a	3.38%
	LGC_HD	n/a	5.41%
Proposed approaches	**MSALBP**	**P=16 , R=2**	**1.35%**
	ELLBP	**N=17,v1=0.7,v2=0.3**	**1.35%**
CASIAMS (S460)			
Reference	**Method**	**Parameters**	**EER**
Jin *et al.* [11]	ILBP	P=8 , R=1	9%
Tao and Veldhuis [125]	SLBP	P=8 , R=1	31%
Wolf *et al.* [12]	TPLBP	w=3, r=2, S=8, α=5 and τ=0.01	31%
	FPLBP	w=3, r1=4, r2=5, S=8, α=1 and τ=0.01	45%
Petpon and Srisuk [126]	LLBP	N=19	5%
	LLBP	N=17	8%
	LLBP	N=15	6%
	LLBP	N=13	6%
Ahmed [127]	GDP	τ=2	6%
Liu *et al.* [116]	ILBPN	N=9	58%
Tong *et al.* [13]	LGC	n/a	22%
	LGC_HD	n/a	22%
Proposed approaches	**MSALBP**	**P=8 , R=2**	**2%**
	ELLBP	**N=7,v1=0.2,v2=0.8**	**0%**

applying fair comparisons, the same resize has been considered for each Region of Interest (ROI). That is, the resize has been normalized to 30 × 150 pixels after implementing any LBP function to a certain ROI. In addition, similar blocking size normalization, feature vector preparation and neural network method have been implemented to all LBP types, as in Table 5.5.

The Receiver Operating Characteristic (ROC) curves for the three databases have been generated by following the novel approach in [27]. The three ROC graphs are depicted in Figs. 5.24, 5.25 and 5.26. The details of producing the ROC curve from the PNN can be found in Appendix 1.

It can be investigated that the ILBP which was suggested to include the centre pixels in the basic LBP attained EER values of 0.79%, 2.70% and 9% for PolyU3D2D, IIT Delhi and CASIAMS (S460) databases respectively as it provides more information. In contrast, the EER values for the SLBP is far from the best performance percentages (more than double) in the PloyU3D2D database and the CASIAMS (S460) database, where they achieved 1.47% and 31% respectively, because theses databases have low FT information regarding to their low

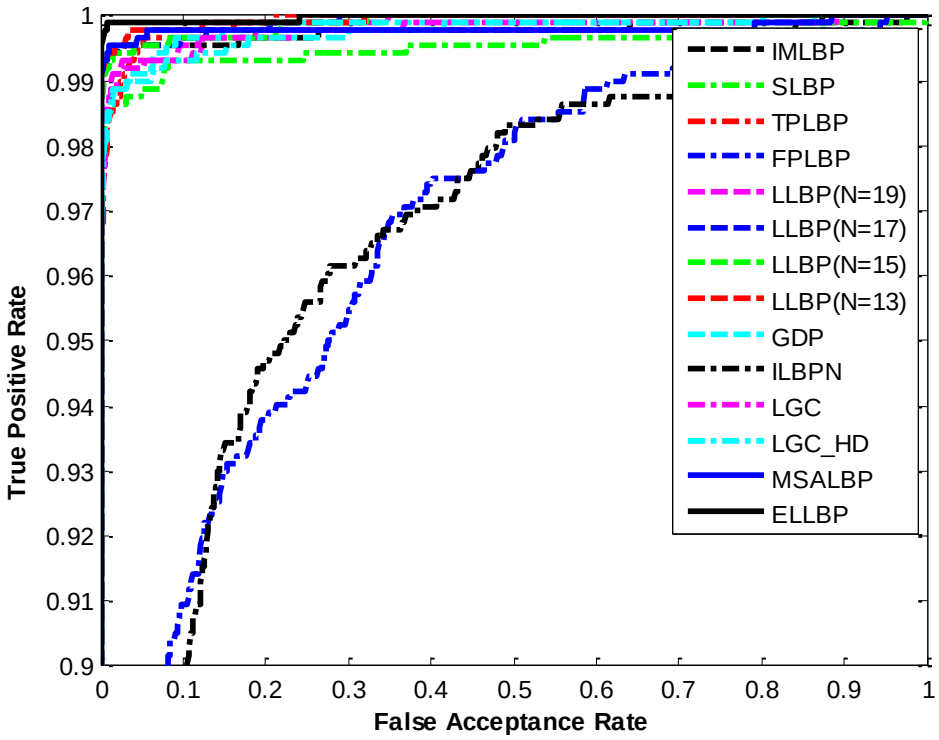


Figure 5.24: ROC curves of various LBP types which have been used in this study for the PolyU3D2D database (The Y-axis scale is reduced to make the figure more clearer)

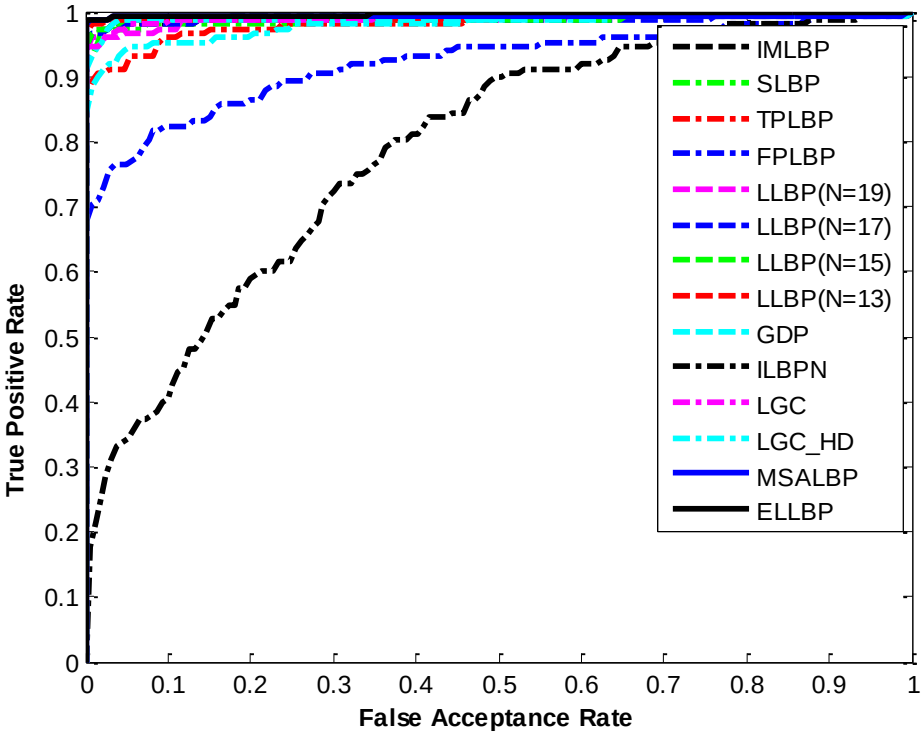


Figure 5.25: ROC curves of various LBP types which have been used in this study for the IIT Delhi database

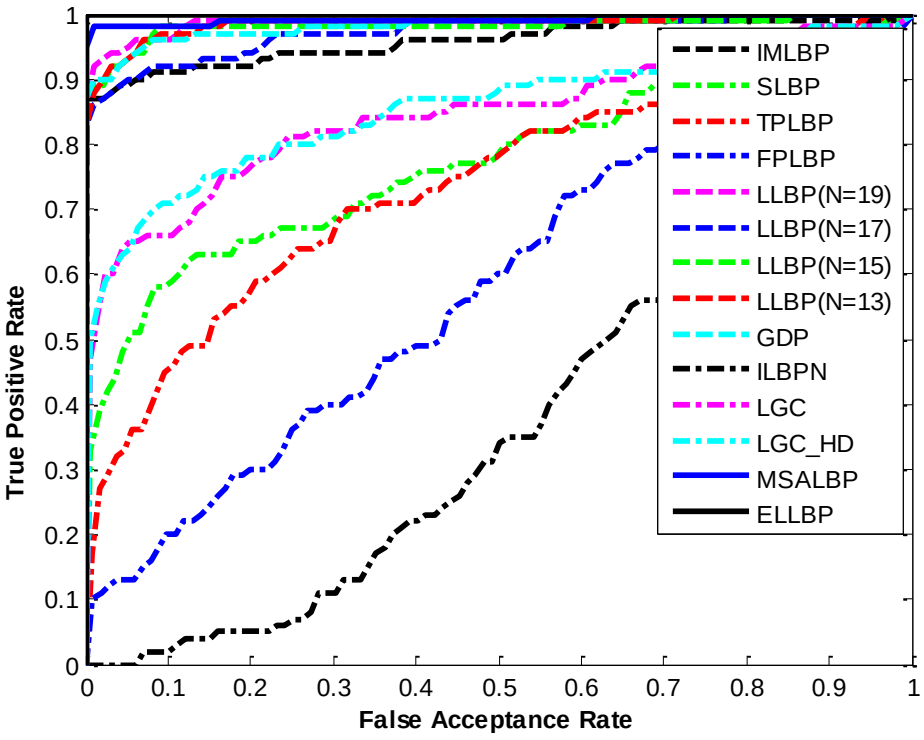


Figure 5.26: ROC curves of various LBP types which have been used in this study for the CASIAMS (S460) database

resolutions. It yielded a value of 2.70%, which is equal to the EER value of the ILBP and twice the best recorded EER in the high resolution IIT Delhi database. This is due to how it is describing the features where the pixel values are between 0-8 and this seems insufficient in calculating the variances between the features. Moreover, the TPLBP and the FPLBP attained very high EERs in almost all the three databases such as 6.76% for the TPLBP and 15.54% for the FPLBP with the IIT Delhi database, and 31% for the TPLBP and 45% for the FPLBP with the CASIAMS (S460) database. The main problem here is that when the operators are applied to the FT images, they lose micro-texture information. This is because TPLBP and FPLBP use patches of sub-blocks instead of pixels to generate their code values. According to their design, they may not capture micro-textures from the images. Also, increasing the radius between the neighbour patches and the center pixel will increase the error rate value and this is why the FPLBP has recorded higher EERs than the TPLBP in all three databases. The LLBP achieved a reasonable performance, due to the fact that it considers the vertical and horizontal lines in its operator and this is appropriate for the main features of the FTs. Since the best lengths of the LLBP vectors are (N=13, N=15, N=17 and N=19) as suggested in [126], all of these lengths have been considered in this work. For example, the LLBP of the lengths N=19, N=17, N=15 and N=13 obtained the EER values of 1.24%, 0.68%, 0.79% and 0.68% respectively for the PolyU3D2D database, and the LLBP of the same ordered lengths achieved 4.05%, 2.70%, 2.70% and 2.03% respectively for the IIT Delhi database. Although the LLBP operators are the closest methods to the ELLBP approach, their results are still far from those achieved with the ELLBP. It can be also noticed that the GDP operator has attained poorer results than the proposed MSALBP and the ELLBP. As mentioned in Section 5.3.6, important textures might be lost in this method because of the tolerance function condition. This method can work reasonably in the high resolution database IIT Delhi where the ERR value achieved 2.70%, but its calculations are influenced by the micro-textures in the low resolution database as in the PolyU3D2D database as the ERR value was equal to 1.36%. Also, this can be observed in the CASIAMS (S460) database as it has obtained a high EER value of 6%. Overall the three databases show that utilizing the ILBPN produces inferior results. That is, 10.17%, 29.05% and 58% for the PolyU3D2D, IIT Delhi and CASIAMS (S460) databases, respectively, are reported. The ILBPN has been

basically designed for the inner knuckle patterns and this could be the reason why it achieved the largest EERs. The LGC and the LGC-HD types which are considered as new types of LBP obtained poorer performances according to their operations of considering the local gradients around the textures. This is expectable as they are taking into account the gradients of the textures around the centre pixel and this is not appropriate for the FT characteristics specifically the diagonal gradients, which have smaller influence than the horizontal and vertical gradients. It is noteworthy that the ERRs of the LGC and the LGC-HD have equal values of 22% in the S460 from the CASIAMS database. As it has been previously mentioned that the images of this database have been acquired under a colour light, which affected the specifications of the hand images. So, the horizontal textures in the FTs have dominated the vertical textures. This can be checked by returning back to the choosing of the $v1$ and $v2$ values in the ELLBP, where the horizontal weight in $v2$ has collected higher value than the vertical weight in $v1$.

It is clear for all three databases that the suggested approaches have generated noticeable improvements in the EER values. The proposed MSALBP feature extraction attained the second best results of 0.68% and 2% in the PolyU3D2D and CASIAMS (S460) databases respectively compared with the ELLBP; whereas, it obtained the best EER of 1.35% in the IIT Delhi database. This confirms the efficiency of this descriptor as it considers the gradient angles of the textures and it uses the determined parameters P and R in its feature extraction, where these parameters concentrate on the most rich features to be processed. On the other hand, the ELLBP method has benchmarked the superior performance of the EERs over all three databases. This is due to its ability to fuse the vertical and horizontal features in collecting the ELLBP codes. In this approach, remarkable values can be observed for different resolution of database images, where in the PolyU3D2D databases has attained 0.34% and this value is obviously a half than that of the second best reported value 0.68% in the MSALBP and in some lengths of the LLBP. Also, in the IIT Delhi database it has produced 1.35% and this EER value is the lowest verification value, which has been recorded in the comparison table. Finally, the ELLBP has achieved a best verification in the CASIAMS (S460) database by producing 0% and this is further evidence about the ability of the ELLBP in terms of FT feature extractions.

The timings of the operators are also investigated in this study. All the recorded computational time were conducted on a 3.2 GHz Intel Core i5 processor with 8 GB of RAM. See Table 5.6 for the timing comparison between the different employed LBP operators.

Table 5.6: Averaged timing comparison between the different LBP operators for a single finger

Reference	Method	Parameters	Time (sec.)
Jin *et al.* [11]	ILBP	P=8 , R=1	0.08
Tao and Veldhuis [125]	SLBP	P=8 , R=1	0.03
Wolf *et al.* [12]	TPLBP	w=3, r=2, S=8, α=5 and τ=0.01	0.007
	FPLBP	w=3, r1=4, r2=5, S=8, α=1 and τ=0.01	0.007
Petpon and Srisuk [126]	LLBP	N=19	0.078
	LLBP	N=17	0.07
	LLBP	N=15	0.063
	LLBP	N=13	0.06
Ahmed [127]	GDP	n/a	0.05
Liu *et al.* [116]	ILBPN	N=9	0.034
Tong *et al.* [13]	LGC	n/a	0.05
	LGC-HD	n/a	0.05
Proposed approaches	**MSALBP**	**P=16 , R=2**	**0.004**
	MSALBP	**P=8 , R=2**	**0.002**
	ELLBP	N=17	0.06
	ELLBP	N=7	0.04

It is appeared from Table 5.6 that the timings vary between some LBP types and are comparable between the others. It depends on the processing time of each function. The ILBP has recorded a higher time than the others as it considered additional information (the center pixels) in its process. The SLBP has reported the reasonable fast processing time; because its operation is simple and it ignored the weights of converting the binary codes to the decimal codes. Both the TPLBP and the FPLBP have shorter time than all others except the MSALBP approach; because in their processing they work with patches of sub-blocks instead of small sub-blocks in pixel levels. The LLBPs of the lengths (19, 17, 15 and 13) are gradually decreased according to the lengths of the processed vectors. The GDP process has reported a comparable time with the ELLBP but slower time than the MSALBP as it has comparisons with a tolerance in its binary transformation condition for each basic sub-block of pixels. The ILBPN has stated approximately

a half time than the LLBP and ELLBP (N=17) times due to using just horizontal vectors in its function compared with using both vertical and horizontal vectors in the LLBP and ELLBP (N=17) computations. Likewise the GDP, the LGC and LGC-HD has recorded the similar time. They work with the basic sub-blocks of pixels as they calculate the differences between specific pixels. Furthermore, it can be noticed that the LGC and LGC-HD have reported the same time as these functions have the similar general steps of computing their codes. In the case of the suggested approaches, the MSALBP has declared the best calculation time as it efficiently calculate the useful information taking into accounts the suitable micro-textures to be analysed and the basic LBP process to be followed. Expectedly, when the P parameter of the MSALBP was equal to 16 the timing has almost required double processing time than when the P parameter was equal to 8; this is because the number of the processed pixels were doubled. The ELLBP with the length of N=7 has obtained faster processing time than the ELLBP with the length of N=17, where this is due to the number of processing pixels in each horizontal and vertical vectors. The ELLBP has reported a comparable time with the LLBP which can be considered as the closest method to the ELLBP. In addition, it has a competitive time with the ILBPN, SLBP, LGC, LGC-HD and GDP and faster time than the ILBP. It is true that the ELLBP has shown a slower time than the FPLBP, TPLBP, however, they have benchmarked far results of EERs from those achieved by the ELLBP.

5.6 Summary

To summarize, this chapter concentrated on feature extraction and one of the most attractive methods in this field is the LBP. So, a brief description of this method was given. After that, an overview of enhanced LBP types was provided such as ILBP, SLBP, TPLBP, FPLBP, LLBP, GDP, ILBPN and LGC. Then, the new suggested approaches were explained. The first approach was for the MSALBP descriptor, which is a combination of multiple processes starting from the Sobel vertical and horizontal operators; filtering the FT image of these operators separately; combining the resulting images into one image of angle direction values; applying the MSLBP to the angle direction image; blocking and calculating the COV for each block. The second approach was for the ELLBP, which is an enhanced version of the LLBP. In

the ELLBP a fusion between the vertical and horizontal vectors was suggested based on the weighted summation rule. Then, similar blocking and statistical analysis of the MSALBP was applied to the ELLBP.

Extensive experiments were performed to evaluate both approaches. Furthermore, many comparisons were adopted with related work, and the two suggested feature extraction methods achieved important performance enhancements in the case of the EERs and timings. In general, the MSALBP was found to be the best feature extraction technique for FT biometric systems which required the fast operating time. Whilst, the ELLBP is the best feature extraction for the FT biometric systems which required precise verification decision.

The next chapter will consider the fusion between fingers in both directions: feature level fusion and score level fusion. Then, missing FT elements will be assessed for both fusion levels. Consequently, a novel method to salvage the missing information will be explained.

Chapter 6

Finger Fusions, Missing Finger Texture Elements and Salvage Approach

6.1 Introduction

After constructing the feature texture vectors, they are used as inputs to an Artificial Neural Network (ANN) for verifying people. Hence, fusions between the feature vectors can enhance the verification performance. Moreover, evaluating the finger fusion methods in the case of finger amputations is worth investigating and this was never mentioned in prior studies. Hereafter, a method to salvage the missing data can be proposed. Thus, four major contributions are considered in this chapter:

- A powerful human verification scheme based on a finger Feature Level Fusion with the Probabilistic Neural Network (FLFPNN) will be described.
- An innovative Finger Contribution Fusion Neural Network (FCFNN) method will be suggested, implemented and evaluated.
- The verification performance will be evaluated for limited Finger Texture (FT) views or even with missing fingers.
- A novel approach will be suggested to enhance the error rates after the missing FTs have been salvaged from the features embedded in the trained FLFPNN and FCFNN.

Firstly, a PNN has been applied as a multi-classifier to perform the verification based on the feature level fusion of the five fingers (thumb, index, middle, ring and little). This FT fusion method is the FLFPNN. Then, a multi-object fusion method approach named FCFNN is designed and explained. This approach is inspired from the contribution of each finger; where the contribution score of any finger in terms of personal verification is not equal to the contribution score of the other fingers. Moreover, experiments have been included to examine the verification performance with missing finger elements. For example, removing a distal phalanx, or a distal and an intermediate phalanxes or even a full finger image. An approach has also been suggested, implemented and analysed to increase the verification performance rates in the case of such missing elements.

The remainder of this chapter is organized as follows: Section 2 describes the general specifications of the ANN. The main procedure of the FT verification fusions will be described in Section 3. Section 4 illustrates the proposed missing FT features together with the salvaging approach. The results, discussions and comparisons will be included in Section 5. Finally, the summary of this chapter will be given in Section 6.

6.2 Artificial Neural Network

An ANN is a famous training subject in artificial intelligence. It can be used efficiently for various applications. In recent years, it has found widespread application in different fields biometric verification, identification and classification are examples [25, 133, 147].

The basic idea of the ANN is simulating a biological neural network to represent a non-linear structure suitable for classification or implementing simple functions. An ANN is usually utilized to address a specific application. Fig. 6.1 demonstrates a biological neural network and a simple neural network [175].

The ANNs are divided according to their architectures into two categories: single-layer and multiple layer ANNs. The words “single” and “multiple” are principally referring to the number of connection layers between the main neuron layers. Examples of a single layer and multiple layers ANNs are given in Figs. 6.2, 6.3 and 6.4.

On the other hand, there are two main types of the ANNs according to the

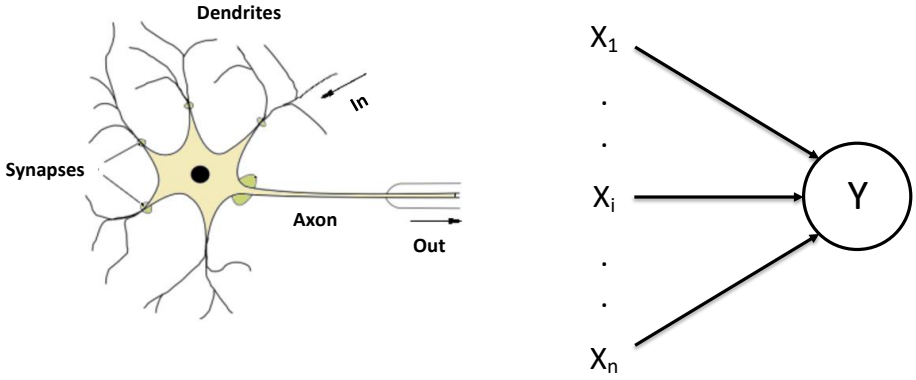


Figure 6.1: A biological neural network and a simple neural network. The biological neural network is shown at the left [14] and a simple neural network is demonstrated at the right

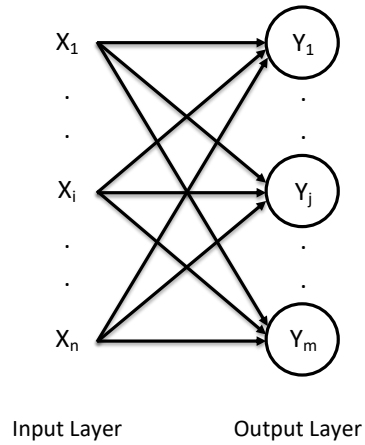


Figure 6.2: A single layer ANN consisting of an input layer, an output layer and a single layer of connections

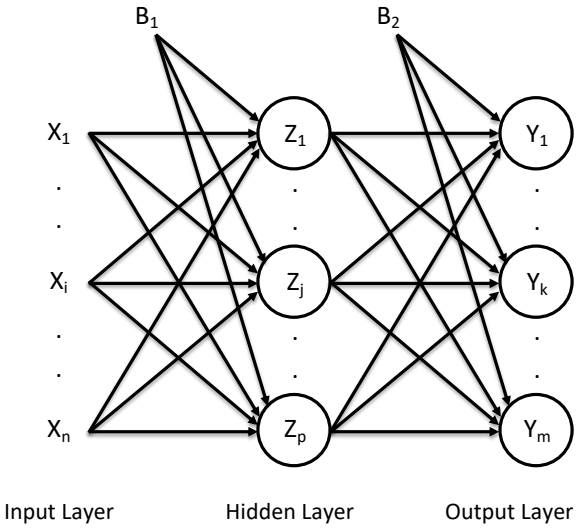


Figure 6.3: A multiple layer ANN which consists of an input layer, a hidden layer, an output layer and two layers of connections

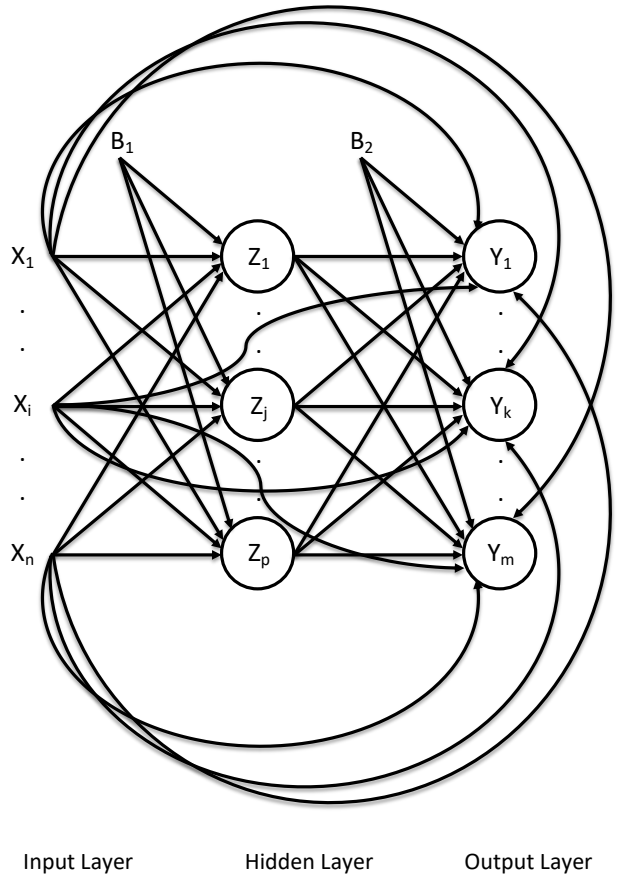


Figure 6.4: A fully-connected multiple layer ANN which consists of an input layer, a hidden layer, an output layer and three layers of connections

training style, supervised training and unsupervised training. Fundamentally, the term "supervised" refers to networks that require targets in training, whilst, unsupervised networks do not require targets. In both cases, the aim of any learning or training is to update the learning weights inside the ANN in order to accomplish a successful relationship between the inputs and the desired or expected outputs. Moreover, there are two essential phases in the ANNs, which are known as the training and the testing phases. During the training phase, the ANN learns by using determined input samples. The learning weights after the training process are stored in order to be applied in the testing phase. Subsequently, during the testing phase the ANN is collecting unseen samples and utilizing the stored weights to construct its output decisions [175].

6.3 Finger Fusions

6.3.1 Feature Level Fusion with the Probabilistic Neural Network

For the case of finger fusion, the FLFPNN method has been used, where a powerful verification structure was exploited and implemented based on the feature level fusion and the PNN technique. This structure is depicted in Fig. 6.5. It starts from collecting a hand image, segmenting the five fingers (thumb, index, middle, ring and little) to separate images and applying a feature extraction process to the Region of Interest (ROI) of each finger image. Hereafter, establishing and collecting a single variance feature vector is applied as a feature level fusion for the fingers of each input hand image. Consequently, a multi-classifier Probabilistic Neural Network (PNN) was used effectively to verify people.

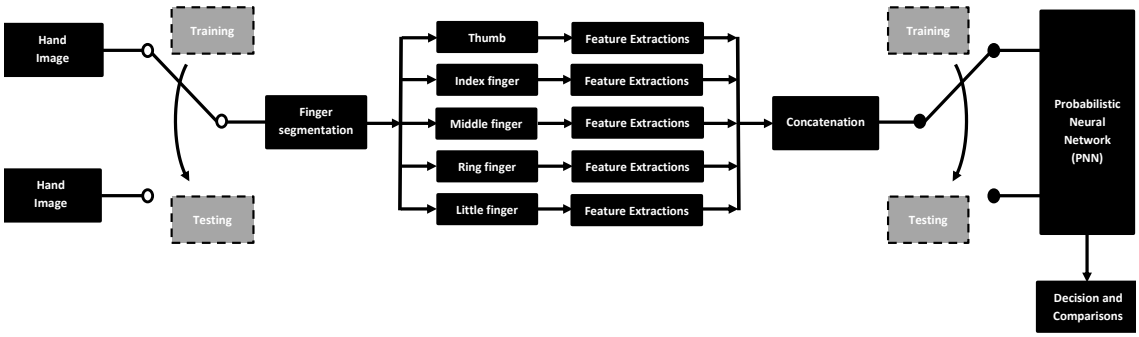


Figure 6.5: The main diagram of the suggested FLFPNN verification system; the switches should be used together to swap between the training and the testing phases

The feature level fusion procedure was described and performed in [176–178]. It consists of three main stages: normalization, selection and concatenation [176]. For the data normalization stage, the resize of each ROI has been appropriately chosen for all fingers to be equal 30×150 after implementing any Local Binary Pattern (LBP) function. Each finger has been divided into non-overlapped square cells each with 5×5 pixels as illustrated in Chapter 5 and following [25]. Henceforth, to extract the FT features and reduce the feature vector size, a statistical computation called the Coefficient of Variance (COV) has been applied to each cell. Subsequently, all the COV values for all five fingers have been concatenated to perform one single vector. Let:

$$\mathbf{F}_{Fing} = \{COV_{1,Fing}\ , COV_{2,Fing}\ , COV_{3,Fing}\ , ...\ , COV_{m,Fing}\} \tag{6.1}$$

where $\mathbf{F}_{Fing}$ represents the variances of the FTs, $Fing$ refers to one of the five fingers ($Fing = 1, 2, ..., 5$), COV denotes the COV value and m is the number of COV values in each finger. By concatenating these vectors, a single 1D vector $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4, \mathbf{F}_5]$ has been established to represent the full features of one sample.

A PNN is considered as a multi-classifier ANN. It is also termed as a multiple layer neural network as it consists of three connection layers distributed between the main neural layers: an input layer, pattern layer, summation layer and decision layer. Principally, the PNN utilizes the following Probability Density Function (PDF) [175, 179]:

$$f_A(\mathbf{x}) = \frac{1}{(2\pi)^{l/2}\sigma^l}\frac{1}{m_A}\sum_{i=1}^{m_A} exp\left[-\frac{(\mathbf{x}-\mathbf{x}_{Ai})^T(\mathbf{x}-\mathbf{x}_{Ai})}{2\sigma^2}\right] \tag{6.2}$$

where $\mathbf{x}_{Ai}$ represents the i^{th} input training pattern from the class A, l represents the dimension of the input vectors, m_A is the number of training patterns in class A, σ is a smoothing parameter for the Gaussian distribution function and $(.)^T$ is the transpose operator. In this work, m_A has been considered to be the same for each class, which is denoted as p, and c denotes the number of classes.

The node outputs of the pattern layer are computed according to the following Equation (6.3) [175, 180], which is also known as a Radial Basis Function (RBF):

$$Z_{i,j} = exp\left[-\frac{(\mathbf{x}-\mathbf{w}_{i,j})^T(\mathbf{x}-\mathbf{w}_{i,j})}{2\sigma^2}\right], \quad i = 1, 2, ..., p\ ,\ j = 1, 2, ..., c \tag{6.3}$$

where $Z_{i,j}$ represents the output of a pattern node, $\mathbf{x}$ represents the input vector $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ and $\mathbf{w}_{i,j}$ is the i^{th} vector of class j, so, the training vector can be represented as $\mathbf{w}_{i,j} = [w_1, w_2, ..., w_n]^T$.

The summation layer will calculate the probabilistic values from the pattern layer for the same input vector to each class by using the following equation:

$$S_j = \frac{1}{p}\sum_{i=1}^{p} Z_{i,j}, \quad j = 1, 2, ..., c \tag{6.4}$$

where S_j represents the summation layer values.

The decision layer will follow a competitive rule called the winner-takes-all rule. This rule can be represented in the following equation (6.5):

$$D_j = \begin{cases} 1 & \text{if } S_j = max \\ 0 & \text{otherwise} \end{cases}, \quad j = 1, 2, ..., c \tag{6.5}$$

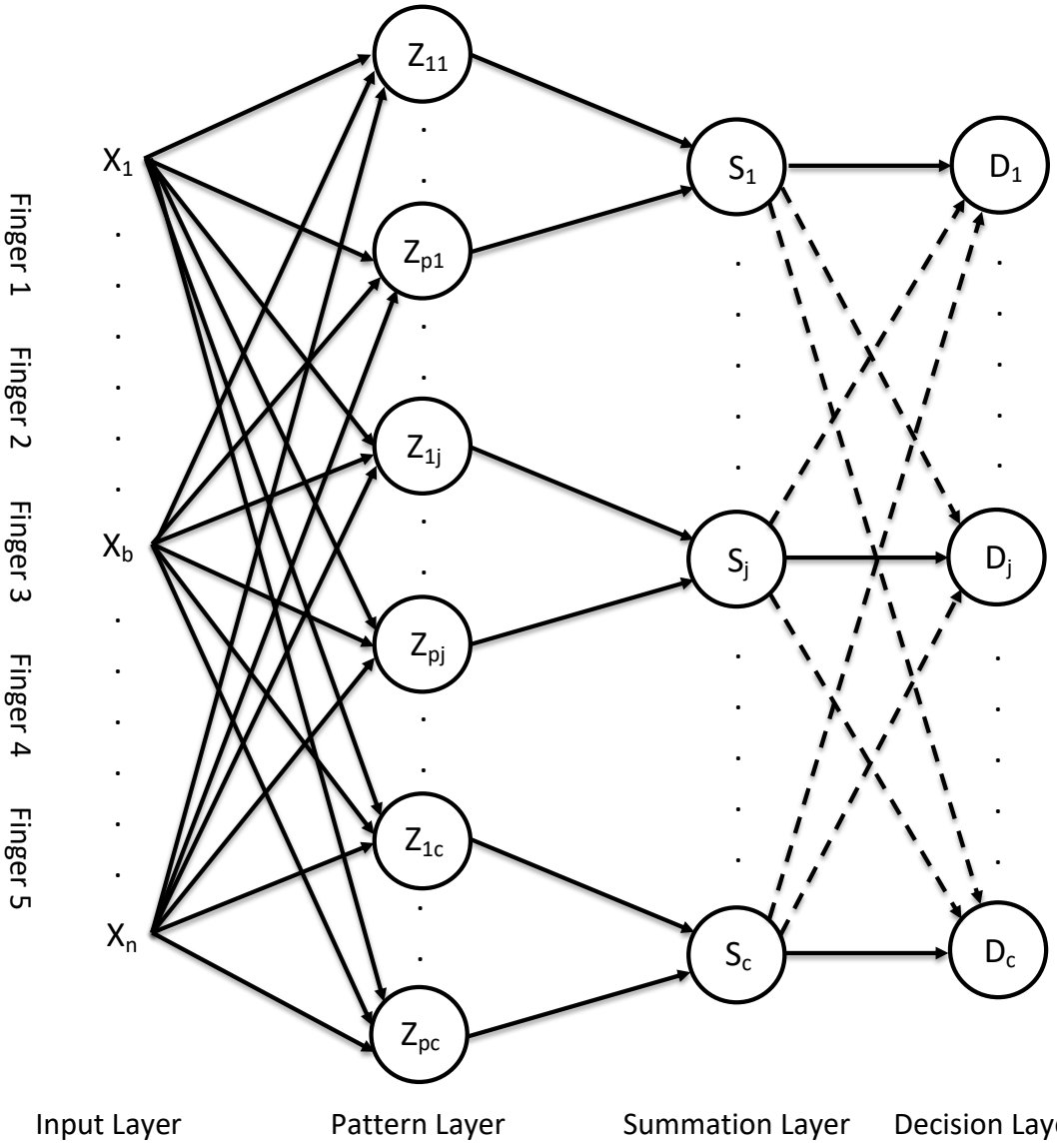


Figure 6.6: The general form of the PNN with the concatenated five finger feature input vector of the five fingers. It includes input, pattern, summation and decision layers

where max represents the extracted maximum S_j value.

Basically, during the training phase the PNN aims to create the weights $\mathbf{w}_{i,j}$ to establish a non-linear relationship between the inputs and the targets. As mentioned, the weights will be stored after completing the training. Then, these weights will be used in the testing phase to predict the outputs according to the input patterns. In the PNN the weight values are generated exactly equal to the training input values. There are some important advantages of using the PNN [52]: it has a very short training time; it does not require more than one iteration; it is not degraded by local minima in the cost function as in the backpropagation neural network; it has a flexible structure, where it is easy to add or remove training data; and one of the most interesting advantages of this network is that the input training patterns will be saved in the connection weights between the input and the pattern layers.

However, it could be argued that this type of ANN has drawbacks as it requires memory and it has slow execution time in the testing phase. But this is not a major problem because of the availability of fast computers with large memories [181].

The general form of the PNN is depicted in Fig. 6.6. In this figure the feature level fusion method based on concatenating the finger feature vectors is used in the input layer.

6.3.2 Proposed Finger Contribution Fusion Neural Network

After analysing the standard PNN, it can be noticed that the information of the input layer will be shared in the output. Therefore, each finger will have the same contribution opportunity if the traditional PNN is used. The finger contribution scores are important due to the fact that each finger has a different contribution according to [117]. From this point it can be argued that the performance of the verification can be enhanced during the testing phase after including the contribution score of each finger. Fig. 6.7 illustrates the skeleton of the proposed fusion method. It consists of: an input layer, pattern layer, contribution layer, summation layer and decision layer. Hence, an extra layer called the "contribution layer" is proposed to be inserted so that different contributions can be acquired from each finger.

This method imposes that each finger should use a separate PNN during the

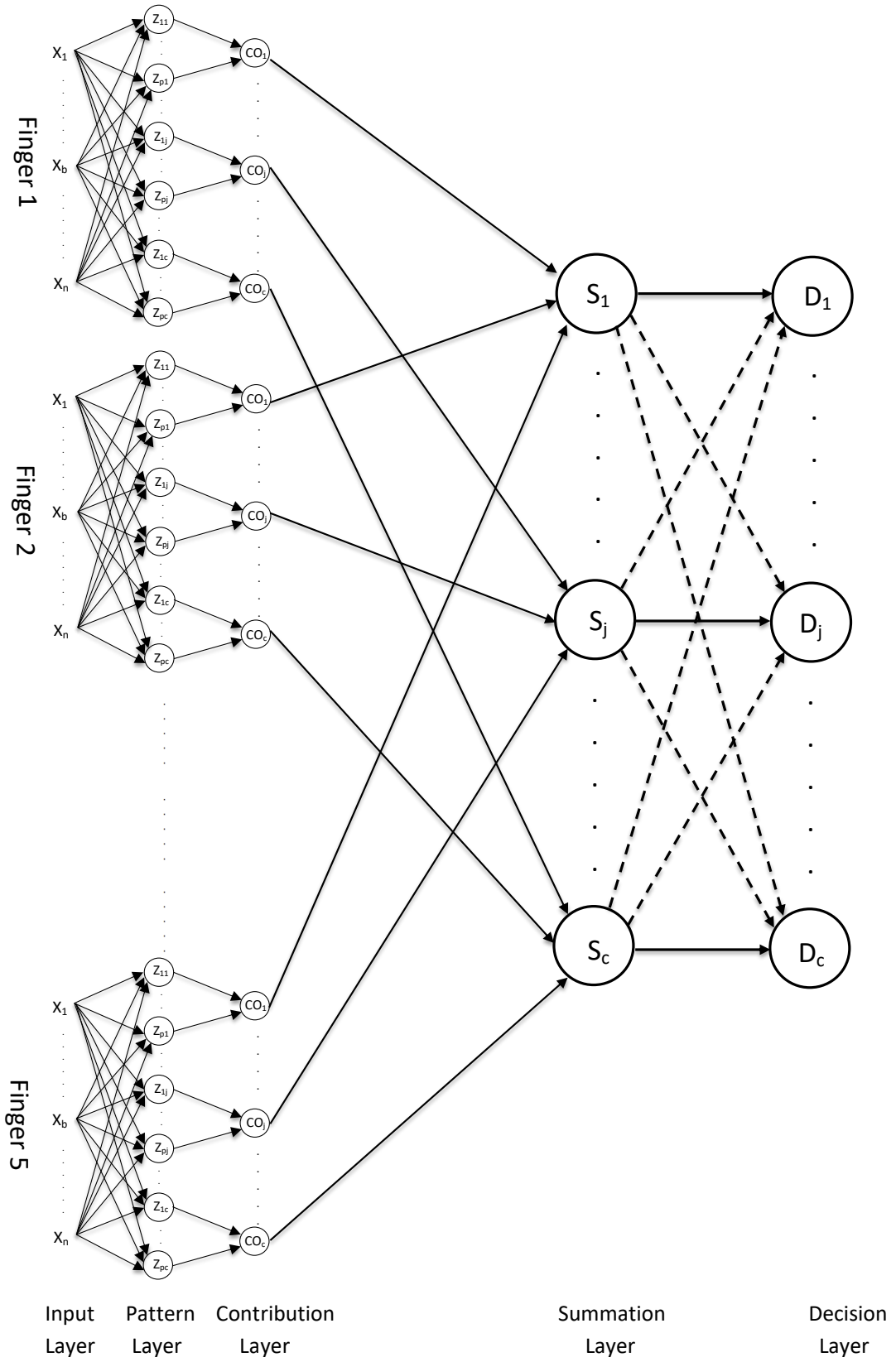


Figure 6.7: The proposed FCFNN method including the contribution layer

training phase to benchmark its contribution scores within the established weights in the pattern layer. On the other hand, during the testing phase a score fusion can be implemented after the contribution layer, where each finger will determine

its contribution score. Next, the contribution score values for the fingers are fused together using the sum fusion rule. After that, the score fusion of all fingers will be fed to the decision layer.

Therefore, Equation (6.3) can be modified as follows:

$$Z_{i,j}^{Fing} = exp\left[-\frac{(\mathbf{x}^{Fing} - \mathbf{w}_{i,j}^{Fing})^T(\mathbf{x}^{Fing} - \mathbf{w}_{i,j}^{Fing})}{2\sigma^2}\right], \quad i = 1, 2, ..., p\ ,\ j = 1, 2, ..., c \tag{6.6}$$

The score values are fused together according to Equations (6.7) and (6.8) as shown below:

$$CO_j^{Fing} = \frac{1}{p}\sum_{i=1}^{p} Z_{i,j}^{Fing}, \quad j = 1, 2, ..., c \tag{6.7}$$

$$S_j = \sum_{Fing=1}^{5} CO_j^{Fing}, \quad j = 1, 2, ..., c \tag{6.8}$$

where $Fing$ represents the finger object and CO_j^{Fing} represents the contribution nodes in the contribution layer.

Nevertheless, Equation (6.5) is still the same in the FCFNN, where the winner-takes-all rule is the basis of the output decision. In the PNN, during the training phase the FCFNN aims to build the $\mathbf{w}_{i,j}^{Fing}$ weights which are stored after training. These weights will be utilized in the same network during the testing phase to examine the input patterns and produce the predicted FCFNN outputs.

Flexibility is one of the main advantages of this proposed method. Thus, if any person accidentally lost any finger it will be easy to remove the connections of the lost finger or only ignore its computations from the FCFNN.

Also, the FCFNN has the same PNN advantages namely: flexibly modifying the number of trainees taking into account that the finger feature vectors do not require the concatenation part to be considered; it does not iterate to establish the matching weights between the inputs and their targets and it is also not degraded by local minima as in the backpropagation network. In addition, it has the ability of saving

the training feature vectors of all input fingers and it computes the distance from the input vectors of the fingers to the stored training patterns during the testing stage. The last specifications can be utilized in salvaging the missing data as will be explained in the next section.

6.4 Missing FTs and salvaging data approach

Missing finger parts have also been investigated in this work, where this can be considered as a new investigation in the case of FT patterns. To the best of the author's knowledge, no publication has previously explored this issue until Al-Nima *et al.* [26]. The missing FT elements have been considered as zero inputs to the neural network according to [175] and according to the black background in the finger segmentation, see Chapter 4:Equation (4.6). Empirically, the first quarter of the ROI represents approximately missing the distal phalanx and the first half ROI represents approximately missing the distal and the intermediate phalanxes in the four fingers (index, middle, ring and little). Whereas, the first third part of the ROI represents approximately missing distal phalanx for the thumb. The verification performance will be expected to be reduced after amputating a part of a finger or even a full finger from the inputs. Therefore, an approach is suggested to salvage the missing elements by taking advantage from the following distance in the RBF Equation (6.3):

$$dist = \left[(\mathbf{x} - \mathbf{w}_{i,j})^T(\mathbf{x} - \mathbf{w}_{i,j})\right], \quad i = 1, 2, ..., p \ , \ j = 1, 2, ..., c \tag{6.9}$$

where $\mathbf{x}$ is the input vector and $\mathbf{w}_{i,j}$ represents the i^{th} vector of class j.
Briefly, each node in the pattern layer in the PNN already saves the full training FT patterns during the training phase. Hence, to describe the salvage approach, the following justification can be described:
Assume: $x'_q = x_b$ when $x_b \neq 0$ $(b = 1, 2, ..., n)$.
Similarly, $w'_{u,v} = w_{i,j}$ when $w_{i,j}$ corresponds to the x'_q $(i = 1, 2, ..., p)$ $(j = 1, 2, ..., c)$.
Also, $dist'$ represents the distance in Equation (6.9) excluding the missing values. They can be represented as shown in Equation (6.10):

$$dist' = \left[(\mathbf{x}' - \mathbf{w}'_{u,v})^T(\mathbf{x}' - \mathbf{w}'_{u,v})\right], \quad u = 1, 2, ..., p \ , \ v = 1, 2, ..., c \tag{6.10}$$

Under the assumptions of clear quality images: the salvaging approach assumes that the stored weight pattern, which can be identified among the different classes according to the minimum $dist'$ value, will generally salvage or rescue the missing data. In other words, $x_b = w_{i,j}$ such that the $dist'$ value is the minimum and $x_b = 0$ for $b = 1, 2, ..., n$.

From this point, one of the three possibilities that can be achieved:

1. The closest pattern of the same class may be confirmed. In this case the missing elements will be salvaged to the same person.

2. The closest pattern may be from a different class, but the probability functions and summations in Equations (6.3) and (6.4) respectively will not define the false class. In this case the salvaged missing elements will be from a different person or class. However, the probability functions of the other weighted patterns from the same class will dominate the wrong decision and it will not be indicated. Instead, the right verification will still be attained.

3. The closest pattern from a different class may be confirmed and the probability functions and summations in Equations (6.3) and (6.4) respectively will verify the false class. In this case the wrong verification decision can be achieved.

The salvage method has an effective performance and will be verified in the next section. It is noteworthy that the proposed salvage approach could be exploited in many biometric applications, where good quality images are observed. A simple example of the salvage process applied to a PNN is given in Figs. 6.8, 6.9, 6.10 and 6.11.

The salvage approach has been modified at the first time in this thesis to be utilized in the FCFNN. As for the PNN, the FCFNN also stores the input training vectors in the connection weights between its pattern and input layers. So, Equation (6.9) can be re-written as Equation (6.11) for the FCFNN:

$$dist_{FC} = \left[(\mathbf{x} - \mathbf{w}_{i,j})^T(\mathbf{x} - \mathbf{w}_{i,j})\right], \quad i = 1, 2, ..., p \ , \ j = 1, 2, ..., c \tag{6.11}$$

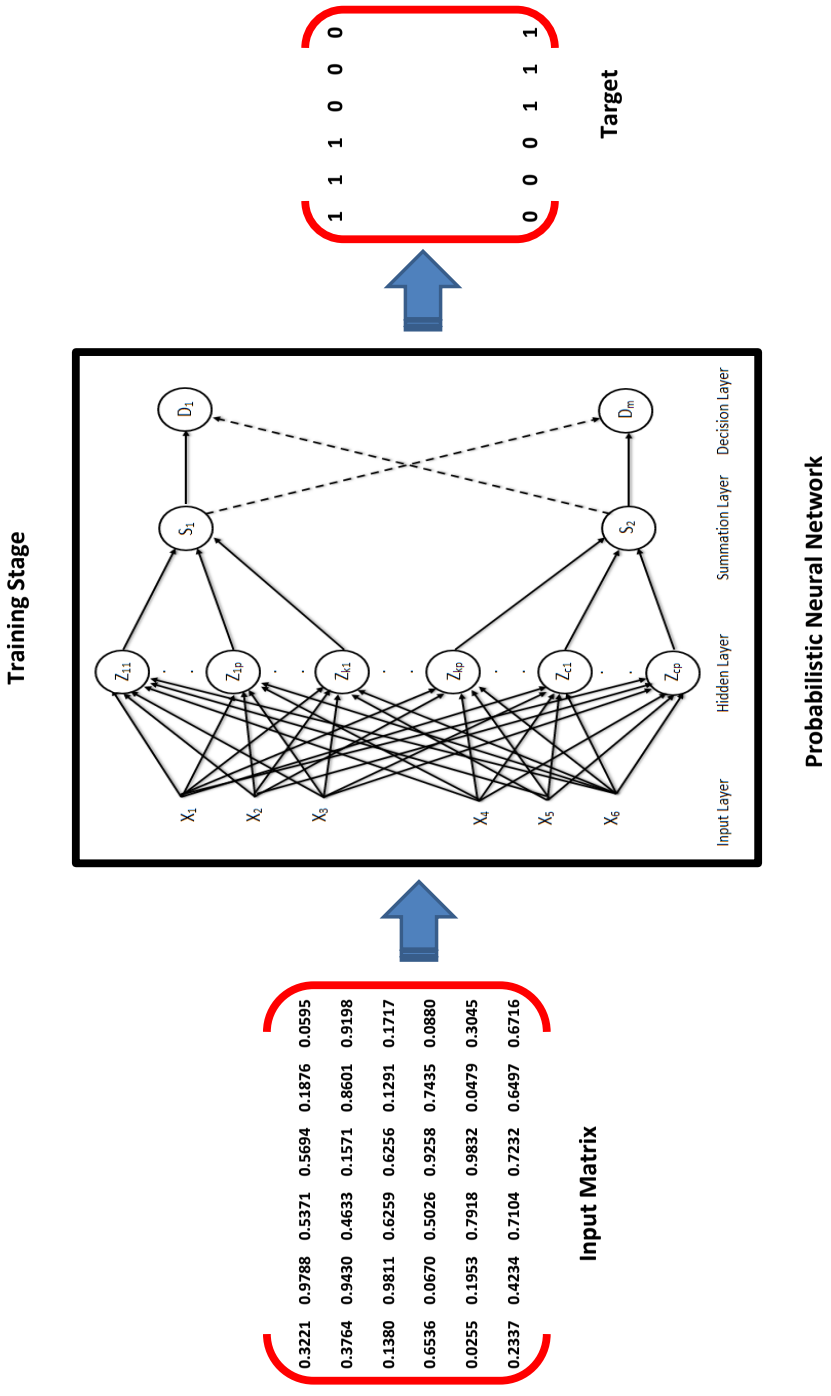


Figure 6.8: Step one of a salvage example clarifying a simple training of a PNN

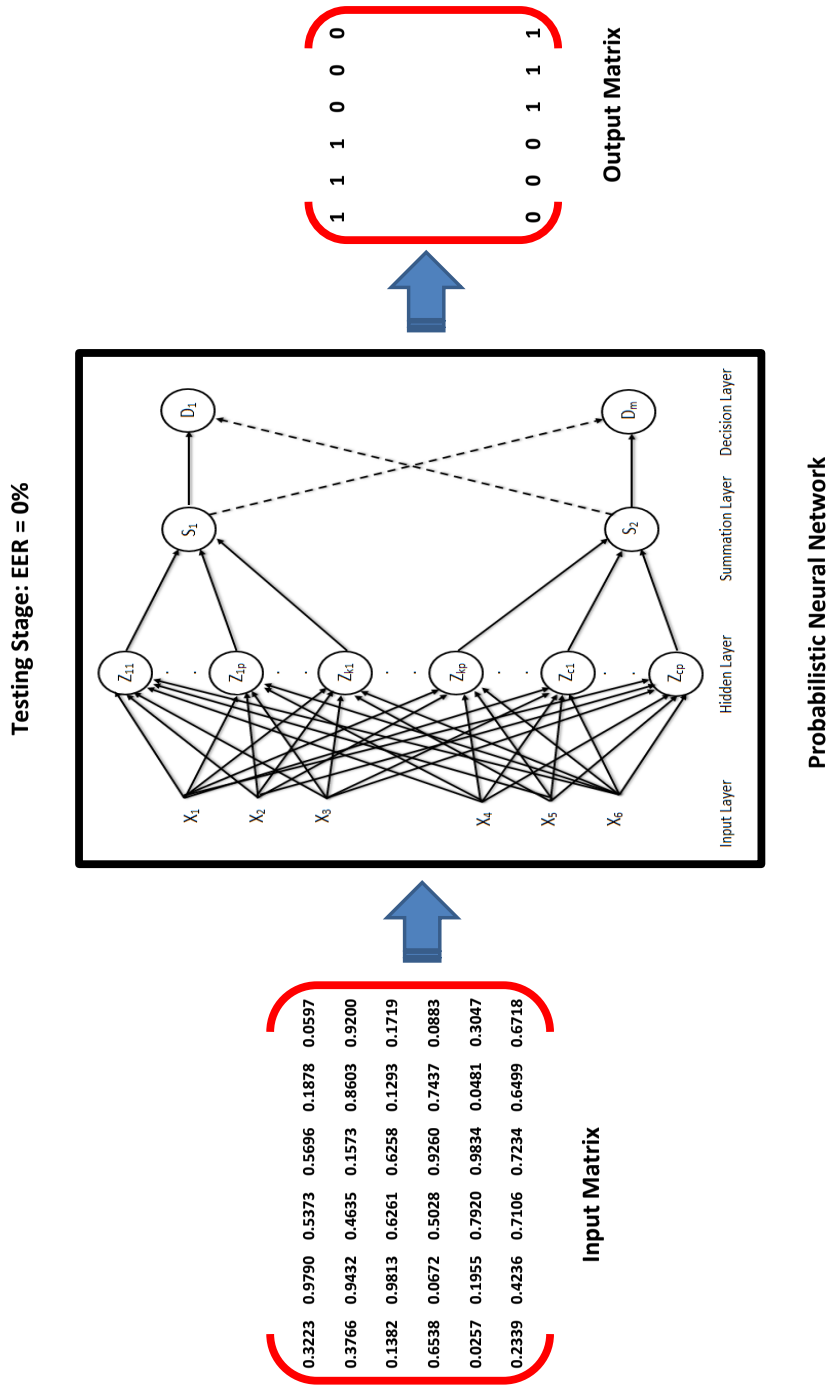


Figure 6.9: Step two of a salvage example clarifying a testing phase, the EER has attained 0%

Figure 6.10: Step three of a salvage example clarifying a testing phase with missing half number of elements, the EER has been increased to 33.33%

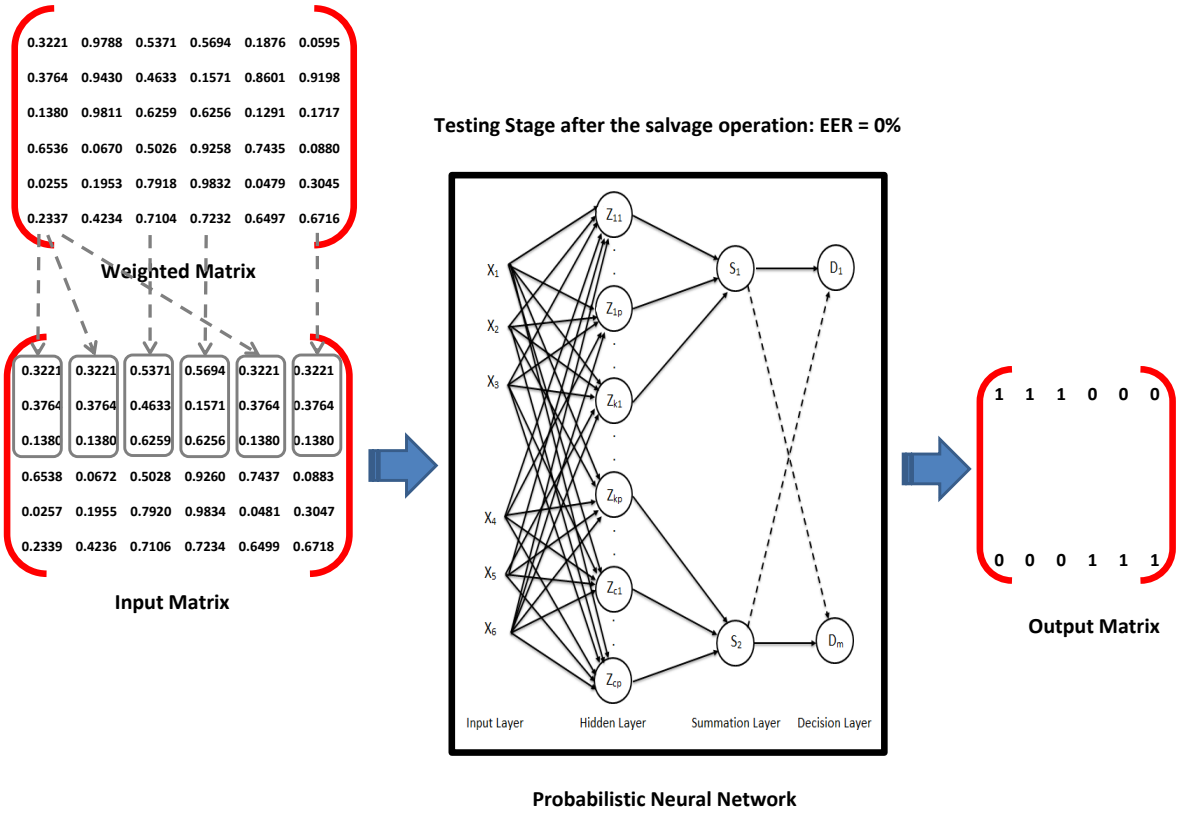


Figure 6.11: Step four of a salvage example clarifying a testing phase with the salvage process, the EER has been decreased to 0%

where $dist_{FC}$ is the calculated distances in the FCFNN, $\mathbf{x}$ is concatenated between the finger input vectors $\mathbf{x}$=[$\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^3$, $\mathbf{x}^4$, $\mathbf{x}^5$] and also $\mathbf{w}_{i,j}$ here is concatenated between the finger weight vectors $\mathbf{w}_{i,j}$=[$\mathbf{w}^1_{i,j}$, $\mathbf{w}^2_{i,j}$, $\mathbf{w}^3_{i,j}$, $\mathbf{w}^4_{i,j}$, $\mathbf{w}^5_{i,j}$]. Furthermore, Equation (6.10) can be modified to be:

$$dist'_{FC} = \left[(\mathbf{x}' - \mathbf{w}'_{u,v})^T(\mathbf{x}' - \mathbf{w}'_{u,v})\right], \quad u = 1, 2, ..., p \ , \ v = 1, 2, ..., c \tag{6.12}$$

where $dist'_{FC}$ is the distance in Equation (6.11) excluding the missing values, $\mathbf{x}'$ is equal to $\mathbf{x}$ but without the missing elements and $\mathbf{w}'$ corresponds to $\mathbf{x}'$.
Similar assumptions, analyses and possibilities can be obtained when using the salvage in this network. However, the only difference is when the contribution layer generates a wrong response and thereby significantly affects the verification decision.

6.5 Results and Discussions

6.5.1 Evaluating the finger fusion methods

The suggested fusion methods, with the FLFPNN and the FCFNN, have been examined and evaluated. Both have two phases, in the first phase their neural networks will be learned for specific training patterns while in the testing phase they will give decisions to the vectors which have not been seen before. Furthermore, they can be considered as multi-object fusions. The FLFPNN structure has been applied in the previous chapters. Comparisons are performed between the FLFPNN and the FCFNN methods in this chapter. The three databases which have been employed are The Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67], Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65, 66] and Spectral 460nm (S460) from the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9]. Table 6.1 shows the differences between the two fusion methods in terms of the EER. The parameters of the feature extractions have been considered according to their best achievements as highlighted in Chapter 5.

Table 6.1: Verification performance comparisons between the FLFPNN and the FCFNN using best feature extraction achievements

PolyU3D2D database			
Feature extraction method	**Finger fusion method**	**Parameters**	**EER**
MSALBP	FLFPNN	P=8 , R=2	0.68%
MSALBP	FCFNN	P=8 , R=2	0.23%
ELLBP	FLFPNN	N=17,v1=0.8,v2=0.2	0.34%
ELLBP	FCFNN	N=17,v1=0.8,v2=0.2	0.11%
IIT Delhi database			
Feature extraction method	**Finger fusion method**	**Parameters**	**EER**
MSALBP	FLFPNN	P=16 , R=2	1.35%
MSALBP	FCFNN	P=16 , R=2	2.03%
ELLBP	FLFPNN	N=17,v1=0.7,v2=0.3	1.35%
ELLBP	FCFNN	N=17,v1=0.7,v2=0.3	1.35%
CASIAMS (S460)			
Feature extraction method	**Finger fusion method**	**Parameters**	**EER**
MSALBP	FLFPNN	P=8 , R=2	2%
MSALBP	FCFNN	P=8 , R=2	2%
ELLBP	FLFPNN	N=7,v1=0.2,v2=0.8	0%
ELLBP	FCFNN	N=7,v1=0.2,v2=0.8	0%

A significant performance has been recorded for the PolyU3D2D database in Table 6.1 after using this proposed FCFNN method, where the verification error rate has been reduced from 0.68% to 0.23% for the Multi-scale Sobel Angles Local Binary Pattern (MSALBP) feature extraction and from 0.34% to 0.11% for the Enhanced Local Line Binary Patterns (ELLBP) feature extraction. Whereas, for the IIT Delhi database a slight increase in the EER value of the MSALBP is observed after applying the FCFNN as it has unnatural located fingers in some hand images as explained in Chapter 3. Therefore, it can be declared that the FCFNN is more sensitive to the input pattern than the FLFPNN. This is because the FCFNN is working based on each finger contribution performance, so, problems in locating a finger in an appropriate form could influence its contribution score. For instance, bending a finger during the image acquisition step. However, it seems that there is no serious drawback of using the proposed FCFNN method as the difference between the errors for the MSALBP is not big and it can be seen that in the same database the ELLBP has obtained the same EER values in both fusion methods. The EERs for the FCFNN with the CASIAMS (S460) database are attained with

the same values in the FLFPNN as these are the best results which could be achieved for both finger fusions and feature extraction methods. Another observation in the results of Table 6.1 is that overall the results of utilizing the ELLBP feature extraction method is better than the results of applying the MSALBP feature extraction. Thus, the ELLBP has confirmed its ability as a feature extraction in attaining the best EER values to obtain the FT specifications comparing with the MSALBP and other tested LBP types, see Chapter 5:Table 5.5.

To produce a comprehensive study, Receiver Operating Characteristic (ROC) graphs have been produced for the PNN in the FLFPNN method as illustrated in the novel approach of [27]. Furthermore, in this thesis generating the ROC graphs has been extended to include the FCFNN method too. The details of establishing the ROC curves from the PNN and from the FCFNN are given in Appendix 1. By utilizing the two proposed feature extractions Figs. 6.12 and 6.13 show the ROC graphs for the PolyU3D2D database, 6.14 and 6.15 show the ROC graphs for the IIT Delhi database, and 6.16 and 6.17 show the ROC graphs for the CASIAMS (S460) database.

It can be reported from the ROC graphs that the FCFNN method in general achieves acceptable performance compared with the FLFPNN. Secondly, the ELLBP feature extraction has attained better results than the MSALBP; this is due to its ability in analysing the vertical and horizontal textures of the FTs.

6.5.2 Evaluating the missing FTs and the salvage approach

This work intends to answer these questions: what will happen in general to the verification performance if a part or full finger is accidentally removed and how the negatively affected verification error can be enhanced?. Each of the three used databases has a specific resolution namely, the IIT Delhi, CASIAMS (S460) and PolyU3D2D have high, low and very low resolutions, respectively. Different features can be observed in each database according to the hand acquisition environment. Tables 6.2 and 6.3 show how missing parts of a finger or even a full finger would affect the verification performance after utilizing the FLFPNN method for the two proposed feature extractions MSALBP and ELLBP. Also, they show the enhancements in the EER values which could be obtained after applying

the salvage approach. Different experiments have been applied in these tables: missing the FTs of the distal phalanx from each finger, missing the FTs of the distal and the intermediate phalanxes from each finger except the thumb and missing the FTs of a whole finger.

To analyse Tables 6.2 and 6.3 in the case of missing a part or full finger for the FLFPNN method, the EER values have, as expected, been increased after enlarging the missing feature area. It appears that missing a distal phalanx will affect slightly the error rate; missing distal and intermediate phalanxes will significantly affect the results and missing a full single finger can lead to the wrong verification decision. Moreover, the proposed salvaging method has confirmed its ability to enhance the verification performance, where dramatic improvements can be found in the tables for the two feature extractions and the three employed databases. Example from Table 6.2, missing the distal thumb in the PolyU3D2D has increased the EER to 1.92% and this percentage has been reduced to 0.68%, which is equal to the best recorded value, after applying the salvage. Similarly, removing the distal thumb in the IIT Delhi and CASIAMS (S460) databases has

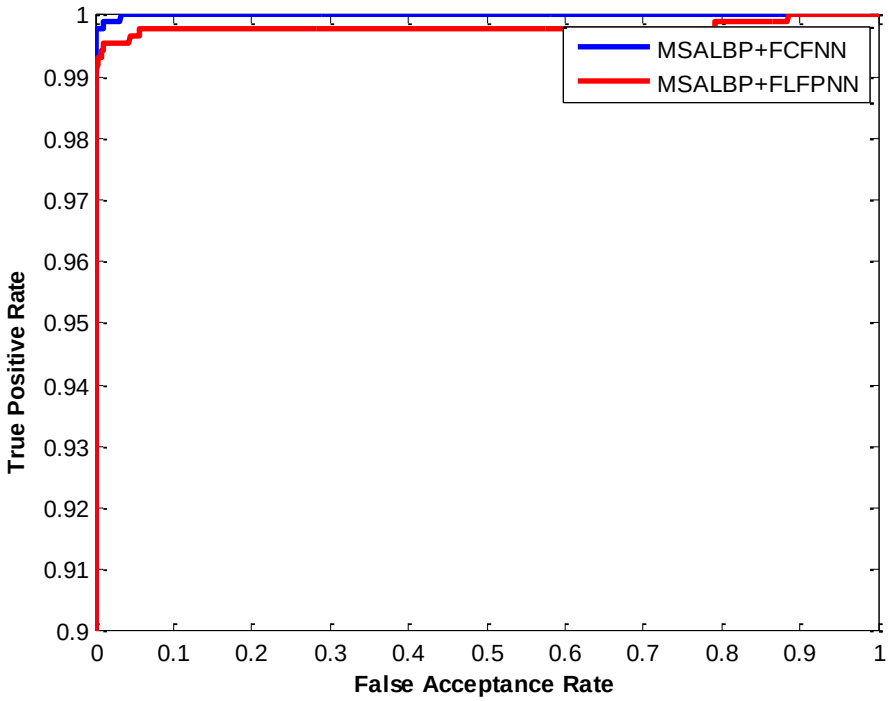


Figure 6.12: The ROC curves of the MSALBP(P=8,R=2) feature extraction with the FLFPNN and for the PolyU3D2D database (The Y-axis scale is reduced to make the figure clearer)

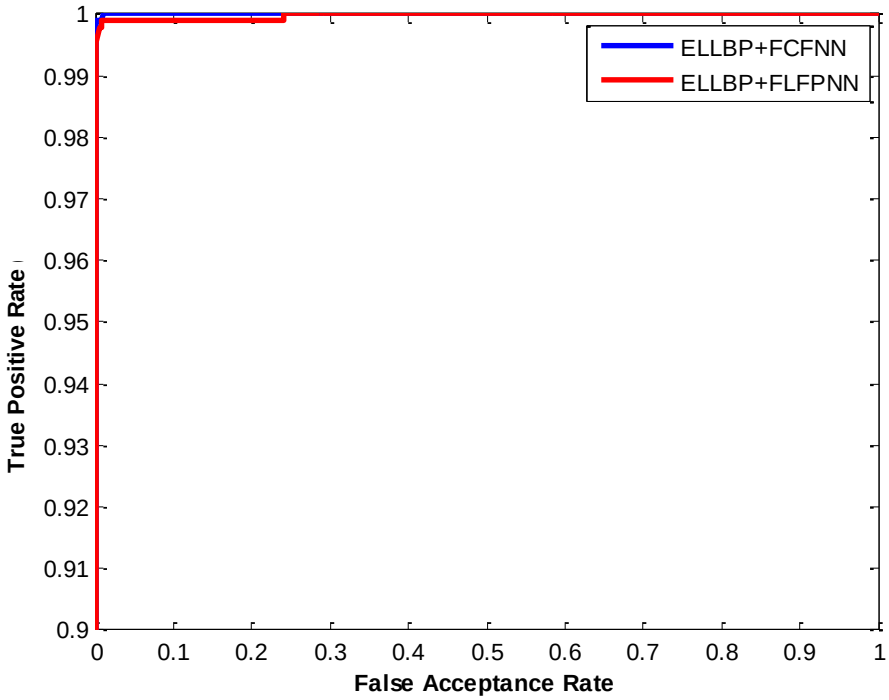


Figure 6.13: The ROC curves of the ELLBP(N=17) feature extraction with the FLFPNN and for the PolyU3D2D database (The Y-axis scale is reduced to make the figure clearer)

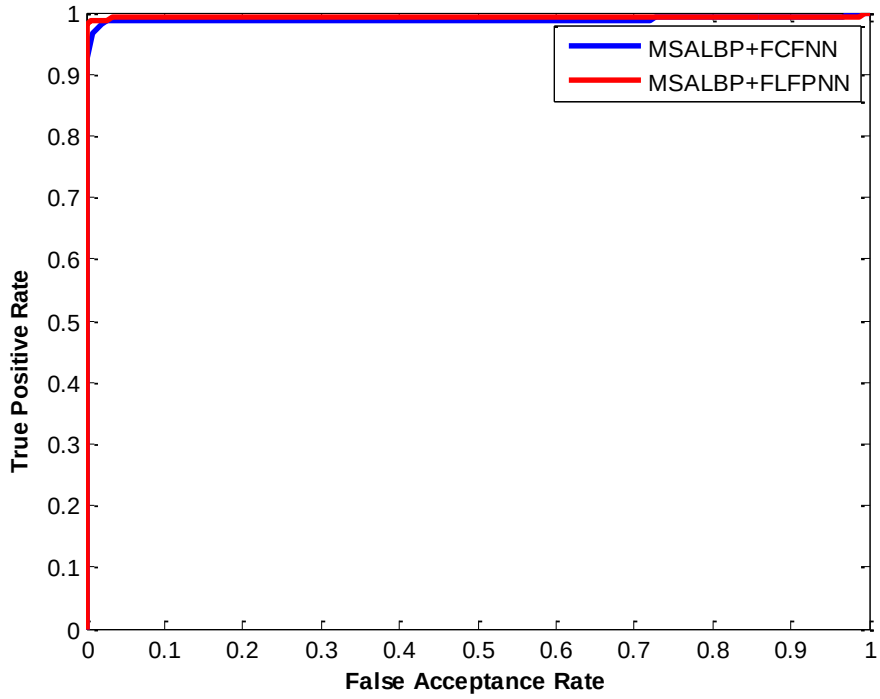


Figure 6.14: The ROC curves of the MSALBP(P=16,R=2) feature extraction with the FLFPNN and for the IIT Delhi database

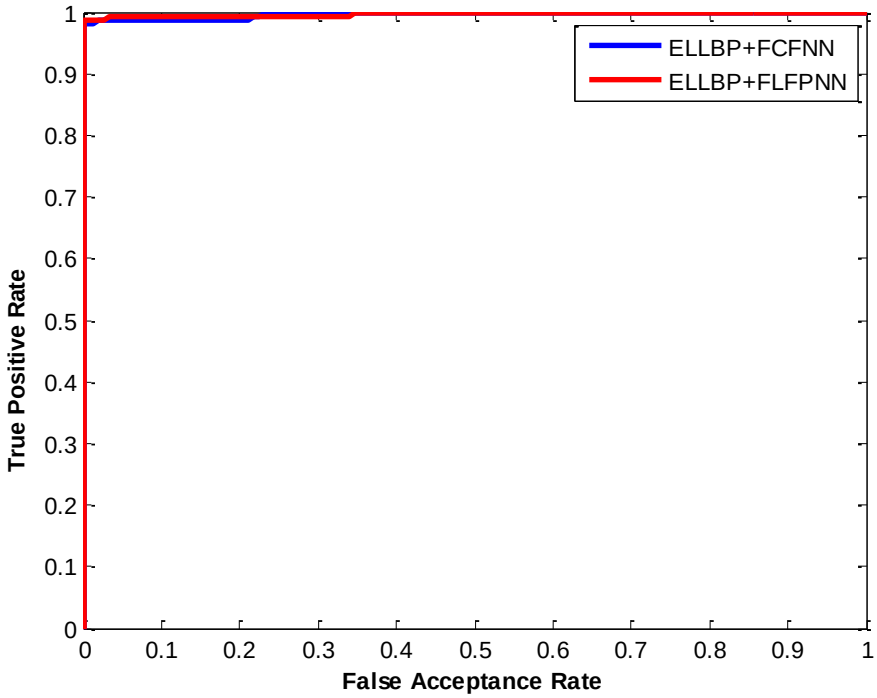


Figure 6.15: The ROC curves of the ELLBP(N=17) feature extraction with the FLFPNN and for the IIT Delhi database

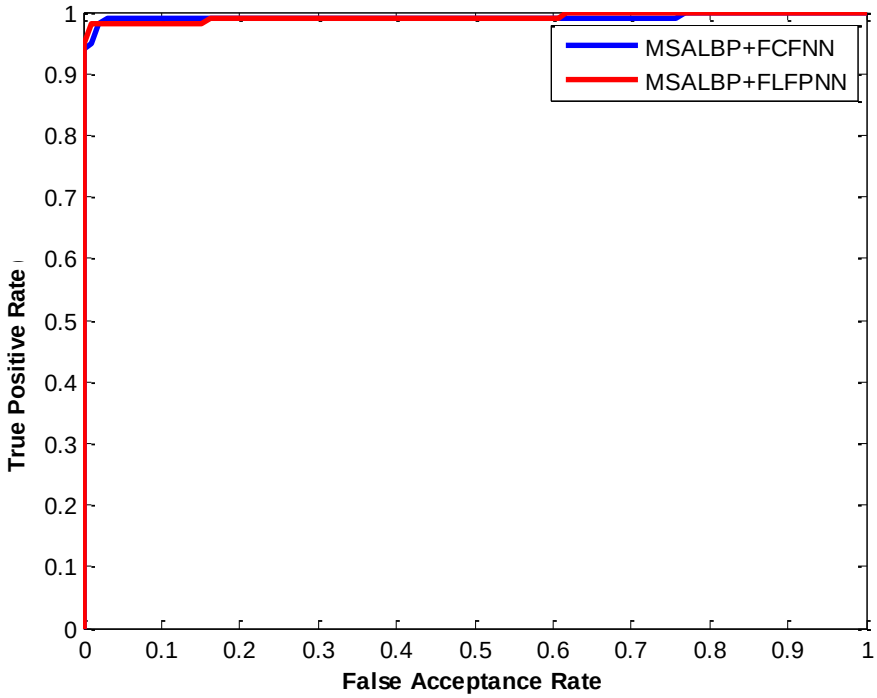


Figure 6.16: The ROC curves of the MSALBP(P=8,R=2) feature extraction with the FLFPNN and for the CASIAMS (S460) database

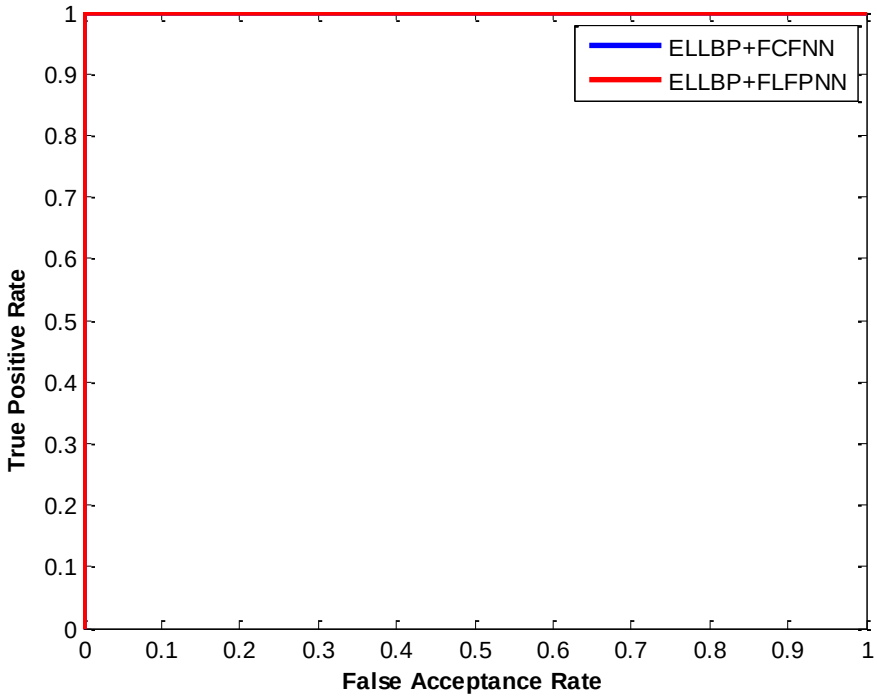


Figure 6.17: The ROC curves of the ELLBP(N=7) feature extraction with the FLFPNN and for the CASIAMS (S460) database

attained 6.08% and 10% respectively before the salvage, and 4.05% and 3% respectively after the salvage. These improvements are increased with more missing FT area. For instance in Table 6.3, removing the distal thumb in the PolyU3D2D database has recorded 0.79% before the salvage and 0.45% after the salvage, so, this has reported an improvement in the percentage change of 43%. Now, comparing this with removing the full thumb in the same database obtained 2.49% before the salvage and 0.57% after the salvage, namely a percentage change of 77%. However, some cases have not been affected by the suggested salvaging method such as missing the distal phalanx from the middle finger in the PolyU3D2D database in Table 6.2 and missing the distal phalanx from the thumb in the IIT Delhi database in Table 6.3, because these regions have not been able to change substantially the verification performance. Furthermore, using the proposed salvage method may affect the results to be slightly better than the best benchmarked EER values in Table 6.1. This is due to the quality of the salvaged part, which might be better than the missing testing part such as salvaging the full little finger in the PolyU3D2D database in Table 6.2, where the EER has achieved 0.45% after the salvage, and salvaging the distal phalanx from the middle finger

Table 6.2: EERs before and after the suggested salvaging method for the MSALBP feature extraction with the FLFPNN

MSALBP(P=8,R=2) of PolyU3D2D database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	1.92%	0.68%
Distal phalanx from the index finger	0.90%	0.68%
Distal phalanx from the middle finger	0.68%	0.68%
Distal phalanx from the ring finger	0.68%	0.68%
Distal phalanx from the little finger	0.68%	0.68%
Distal and intermediate phalanxes from the index finger	1.13%	0.79%
Distal and intermediate phalanxes from the middle finger	1.02%	0.68%
Distal and intermediate phalanxes from the ring finger	1.47%	0.68%
Distal and intermediate phalanxes from the little finger	1.02%	0.68%
Thumb	23.62%	0.68%
Index finger	7.57%	1.02%
Middle finger	5.42%	0.79%
Ring finger	4.07%	0.79%
Little finger	4.18%	0.45%
MSALBP(P=16,R=2) of IIT Delhi database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	6.08%	4.05%
Distal phalanx from the index finger	4.05%	2.70%
Distal phalanx from the middle finger	4.05%	2.03%
Distal phalanx from the ring finger	4.05%	2.70%
Distal phalanx from the little finger	3.38%	2.03%
Distal and intermediate phalanxes from the index finger	5.41%	4.73%
Distal and intermediate phalanxes from the middle finger	6.76%	2.70%
Distal and intermediate phalanxes from the ring finger	8.78%	2.70%
Distal and intermediate phalanxes from the little finger	8.78%	2.70%
Thumb	35.14%*	2.70%
Index finger	18.24%	4.05%
Middle finger	22.30%	2.70%
Ring finger	44.59%	4.05%
Little finger	52.70%	2.70%
MSALBP(P=8,R=2) of CASIAMS (S460) database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	10%	3%
Distal phalanx from the index finger	5%	5%
Distal phalanx from the middle finger	5%	3%
Distal phalanx from the ring finger	2%	2%
Distal phalanx from the little finger	5%	3%
Distal and intermediate phalanxes from the index finger	12%	6%
Distal and intermediate phalanxes from the middle finger	14%	2%
Distal and intermediate phalanxes from the ring finger	8%	2%
Distal and intermediate phalanxes from the little finger	11%	4%
Thumb	69%	5%
Index finger	42%	7%
Middle finger	27%	5%
Ring finger	26%	4%
Little finger	22%	7%

* ROIs of a thumb have been manually corrected for a user who bended this finger during capturing some samples of his/her hand images.

Table 6.3: EERs before and after the suggested salvaging method for the ELLBP feature extraction with the FLFPNN

ELLBP(N=17) of PolyU3D2D database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	0.79%	0.45%
Distal phalanx from the index finger	0.45%	0.34%
Distal phalanx from the middle finger	0.45%	0.23%
Distal phalanx from the ring finger	0.45%	0.34%
Distal phalanx from the little finger	0.68%	0.34%
Distal and intermediate phalanxes from the index finger	0.68%	0.45%
Distal and intermediate phalanxes from the middle finger	0.68%	0.34%
Distal and intermediate phalanxes from the ring finger	1.24%	0.45%
Distal and intermediate phalanxes from the little finger	1.24%	0.34%
Thumb	2.49%	0.57%
Index finger	3.16%	0.68%
Middle finger	3.62%	0.45%
Ring finger	3.28%	0.45%
Little finger	2.82%	0.45%
ELLBP(N=17) of IIT Delhi database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	2.03%	2.03%
Distal phalanx from the index finger	2.70%	1.35%
Distal phalanx from the middle finger	3.38%	1.35%
Distal phalanx from the ring finger	2.70%	2.03%
Distal phalanx from the little finger	1.35%	1.35%
Distal and intermediate phalanxes from the index finger	4.73%	1.35%
Distal and intermediate phalanxes from the middle finger	2.03%	2.03%
Distal and intermediate phalanxes from the ring finger	3.38%	2.70%
Distal and intermediate phalanxes from the little finger	3.38%	1.35%
Thumb	4.73%*	3.38%
Index finger	8.78%	1.35%
Middle finger	8.78%	1.35%
Ring finger	8.78%	2.70%
Little finger	10.14%	2.03%
ELLBP(N=7) of CASIAMS (S460) database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	7%	1%
Distal phalanx from the index finger	4%	0%
Distal phalanx from the middle finger	2%	0%
Distal phalanx from the ring finger	1%	0%
Distal phalanx from the little finger	5%	0%
Distal and intermediate phalanxes from the index finger	10%	1%
Distal and intermediate phalanxes from the middle finger	7%	1%
Distal and intermediate phalanxes from the ring finger	10%	2%
Distal and intermediate phalanxes from the little finger	13%	1%
Thumb	60%	1%
Index finger	43%	3%
Middle finger	41%	1%
Ring finger	63%	4%
Little finger	40%	1%

* ROIs of a thumb have been manually corrected for a user who bended this finger during capturing some samples of his/her hand images.

for the same database in Table 6.3, where the EER has obtained 0.23% after the salvage.

It is worth mentioning that the FLFPNN is influenced by the values of the feature extractions, which are used as inputs to the PNN. After analysing Tables 6.2 and 6.3, it can be noticed that the ELLBP feature extraction is more robust against the illuminations than the MSALBP feature extraction. To clarify, missing one of the terminal fingers (thumb or little finger) significantly affects the verification performance in the MSALBP, where the lighting is located closer to one of these end fingers than the remaining four fingers. Fig. 6.18 shows the illumination distribution in hand images according to the acquisition environment for the PolyU3D2D database.

Figure 6.18: Hand images in the PolyU3D2D database show the illumination distribution and the thumb appears to be brighter than the remaining fingers because of the acquisition lightning environment

From Table 6.2 the EER value after missing the thumb in the PolyU3D2D database is 23.62% and it shows a big difference from the EER of missing any other fingers; the EER value after missing the little finger because of the hand alignment issue in the IIT Delhi database attains 52.70% and it gives a big difference from the EER of missing any other fingers, and the EER value after missing the thumb in the CASIAMS (S460) database is 69% and again this is a big difference from the EER of missing any other fingers. The reason for this is that the MSALBP feature extraction depends on the horizontal and vertical edges which can be detected by using the Sobel method. Therefore, in appropriate degrees increasing the brightness to any finger will reveal more details and more edges can be detected, which is important in the case of missing FTs. On the other hand, these differences are not observed in Table 6.3 as the EER values of using the ELLBP feature extraction show reduced variation between the verification results after removing any finger. This is because the ELLBP is more

robust than the MSALBP in terms of the illumination variances.

An additional point which is worth highlighting is that the EER percentages of the missing FT elements by using the ELLBP in the CASIAMS (S460) database are generally higher than the EER percentages of the missing finger elements in the two other databases such as in missing the thumb where 60% is recorded for the CASIAMS (S460), 4.73% for the IIT Delhi database and 2.49% for the PolyU3D2D database. The reason for this is that the length of the ELLBP in the CASIAMS (s460) database is equal 7, while in the other two databases is equal to 17. So, when the length of the ELLBP is short it will generate small values and this will cause small variances then large EER verification values. On the other hand, when the length of the ELLBP is considerably increased it will generate large values and this will cause large variances then small EER verification values. Now, to examine the FCFNN method in the same cases of missing a part or full finger then applying the salvage approach, Tables 6.4 and 6.5 are fairly established.

These tables show that the EER values in the FCFNN for the missing elements before using the salvage are generally better than the EER values in the FLFPNN for the same corresponding cases. In addition, it can be noticed that the contribution between the fingers in the FCFNN could maintain the results, where missing a small part or a large part of the FTs from a single finger would obtain close EER results for the same corresponding finger. The verification performance might be enhanced after using the salvage process. However, the effects of the salvage are more obvious in missing small FT areas than the large areas. In other words, utilizing the salvage approach in the FCFNN is different from using it in the FLFPNN. To explain that, in the FCFNN when the missing area is increased the enhancement of the EER values is generally reduced and this can be found clearly by using the ELLBP feature extraction. As an instance from Table 6.5, missing the distal phalanx from the ring finger in the CASIAMS (S460) database has recorded 4% and 0% before and after the salvage respectively; missing the distal and intermediate phalanxes from the same finger has obtained 4% and 2% before and after the salvage respectively, and missing the full ring finger has attained 4% and 4% before and after the salvage respectively. By computing the improvements in the percentage change the following results can be noticed: 100%, 50% and 0% for missing the distal, the distal and intermediate and the full ring finger respectively. In contrast, this observation is reversed in the FLFPNN as

Table 6.4: EERs before and after the suggested salvaging method for the MSALBP feature extraction with the FCFNN

MSALBP(P=8,R=2) of PolyU3D2D database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	0.34%	0.45%
Distal phalanx from the index finger	0.68%	0.57%
Distal phalanx from the middle finger	0.68%	0.34%
Distal phalanx from the ring finger	0.45%	0.11%
Distal phalanx from the little finger	0.34%	0.23%
Distal and intermediate phalanxes from the index finger	0.68%	0.57%
Distal and intermediate phalanxes from the middle finger	0.68%	0.34%
Distal and intermediate phalanxes from the ring finger	0.45%	0.45%
Distal and intermediate phalanxes from the little finger	0.34%	0.34%
Thumb	0.34%	0.68%
Index finger	0.68%	0.68%
Middle finger	0.68%	0.45%
Ring finger	0.45%	0.45%
Little finger	0.34%	0.45%
MSALBP(P=16,R=2) of IIT Delhi database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	2.70%	2.03%
Distal phalanx from the index finger	2.03%	2.70%
Distal phalanx from the middle finger	2.03%	1.35%
Distal phalanx from the ring finger	2.70%	2.03%
Distal phalanx from the little finger	2.70%	1.35%
Distal and intermediate phalanxes from the index finger	2.03%	3.38%
Distal and intermediate phalanxes from the middle finger	2.03%	3.38%
Distal and intermediate phalanxes from the ring finger	2.70%	2.03%
Distal and intermediate phalanxes from the little finger	2.70%	2.70%
Thumb	2.70%*	3.38%
Index finger	2.03%	2.70%
Middle finger	2.03%	2.03%
Ring finger	2.70%	2.70%
Little finger	2.70%	4.05%
MSALBP(P=8,R=2) of CASIAMS (S460) database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	2%	3%
Distal phalanx from the index finger	5%	4%
Distal phalanx from the middle finger	5%	3%
Distal phalanx from the ring finger	2%	2%
Distal phalanx from the little finger	5%	3%
Distal and intermediate phalanxes from the index finger	5%	6%
Distal and intermediate phalanxes from the middle finger	5%	2%
Distal and intermediate phalanxes from the ring finger	3%	2%
Distal and intermediate phalanxes from the little finger	6%	4%
Thumb	2%	5%
Index finger	5%	7%
Middle finger	5%	5%
Ring finger	3%	4%
Little finger	6%	7%

* ROIs of a thumb have been manually corrected for a user who bended this finger during capturing some samples of his/her hand images.

Table 6.5: EERs before and after the suggested salvaging method for the ELLBP feature extraction with the FCFNN

ELLBP(N=17) of PolyU3D2D database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	0.34%	0.34%
Distal phalanx from the index finger	0.34%	0.11%
Distal phalanx from the middle finger	0.23%	0%
Distal phalanx from the ring finger	0.23%	0.11%
Distal phalanx from the little finger	0.45%	0.11%
Distal and intermediate phalanxes from the index finger	0.45%	0.45%
Distal and intermediate phalanxes from the middle finger	0.34%	0.23%
Distal and intermediate phalanxes from the ring finger	0.23%	0.34%
Distal and intermediate phalanxes from the little finger	0.45%	0.23%
Thumb	0.34%	0.57%
Index finger	0.45%	0.68%
Middle finger	0.34%	0.45%
Ring finger	0.23%	0.45%
Little finger	0.45%	0.45%
ELLBP(N=17) of IIT Delhi database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	2.70%	2.03%
Distal phalanx from the index finger	1.35%	1.35%
Distal phalanx from the middle finger	2.03%	1.35%
Distal phalanx from the ring finger	2.70%	1.35%
Distal phalanx from the little finger	4.05%	1.35%
Distal and intermediate phalanxes from the index finger	1.35%	1.35%
Distal and intermediate phalanxes from the middle finger	2.03%	1.35%
Distal and intermediate phalanxes from the ring finger	2.70%	2.03%
Distal and intermediate phalanxes from the little finger	4.05%	1.35%
Thumb	2.70%*	3.38%
Index finger	1.35%	1.35%
Middle finger	2.03%	1.35%
Ring finger	2.70%	2.70%
Little finger	4.05%	2.03%
ELLBP(N=7) of CASIAMS (S460) database		
The missing element	**EER**	**EER after salvaging the missing features**
Distal phalanx from the thumb	2%	1%
Distal phalanx from the index finger	2%	0%
Distal phalanx from the middle finger	1%	0%
Distal phalanx from the ring finger	4%	0%
Distal phalanx from the little finger	3%	0%
Distal and intermediate phalanxes from the index finger	2%	1%
Distal and intermediate phalanxes from the middle finger	1%	1%
Distal and intermediate phalanxes from the ring finger	4%	2%
Distal and intermediate phalanxes from the little finger	3%	1%
Thumb	2%	1%
Index finger	2%	3%
Middle finger	1%	1%
Ring finger	4%	4%
Little finger	3%	1%

* ROIs of a thumb have been manually corrected for a user who bended this finger during capturing some samples of his/her hand images.

illustrated earlier. Also, it was observed that some EER values have recorded no improvements after using the salvage such as missing the distal and intermediate phalanxes from the ring finger in the PolyU3D2D database in Table 6.4 and missing the distal phalanx from the thumb in the same database in Table 6.5. As in the FLFPNN, the missing regions in these situations could not effectively influence the EER values. Missing full FTs of a single finger is more likely to achieve a worse result after the salvage than before using it in the FCFNN. This is because when a wrong salvaged vector is selected by the salvage process it could be more effective according to the finger contribution concept.

So, it can be analysed that the FCFNN is sensitive to missing small areas. This is because of its structure, where each finger occupies an effective part within this network. Thus, when a small part of this area is missing such as a finger distal, this can lead to producing wrong computations in Equations (6.6), (6.7) and (6.8) and its verification situation would still continue even if the missing area is enlarged when the same wrong computations are generated. Other cases of increasing slightly EER values can be recognised as the missing area is expanded. Examples of these are missing small or large part (the distal or the distal and intermediate) from the ring finger in the CASIAMS (S460) database in Table 6.4 and missing small or large part (the distal or the distal and intermediate) from the index finger in the PolyU3D2D database in Table 6.5.

In other words, there is a recognisable relationship between the missing area and the employed fusion methods. In the FLFPNN, the EER values are dramatically increased when the missing feature area is increased. Whilst in the FCFNN, the EER values might be maintained to their values of missing a small area of a single finger when the missing area is enlarged.

The effects of the salvage approach have also influenced by the missing area. That is, the EER values have been enhanced slightly or have not been enhanced after expanding the missing area by applying the FCFNN as it can be investigated in Tables 6.4 and 6.5. This is different from what has been observed in the FLFPNN in Tables 6.2 and 6.3, where the EER values have been enhanced dramatically after enlarging the missing area.

Although there is a certain drawback associated with the using of the FCFNN regarding to the size of the missing FT area, the additional FCFNN ability of removing the missing finger or fingers from its architecture can be considered as

Table 6.6: Removing a finger or fingers from the FCFNN using the MSALBP feature extraction

MSALBP(P=8,R=2) of PolyU3D2D database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	0.34%
Thumb+Middle+Ring+Little	Index	0.68%
Thumb+Index+Ring+Little	Middle	0.68%
Thumb+Index+Middle+Little	Ring	0.45%
Thumb+Index+Middle+Ring	Little	0.34%
Middle+Ring+Little	Thumb+index	1.24%
Thumb+Ring+Little	Index+Middle	1.58%
Thumb+Index+Little	Middle+Ring	0.57%
Thumb+Index+Middle	Ring+Little	0.90%
Ring+Little	Thumb+Index+Middle	2.71%
Thumb+Little	Index+Middle+Ring	2.71%
Thumb+Index	Middle+Ring+Little	1.58%
MSALBP(P=16,R=2) of IIT Delhi database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	2.70%
Thumb+Middle+Ring+Little	Index	2.03%
Thumb+Index+Ring+Little	Middle	2.03%
Thumb+Index+Middle+Little	Ring	2.70%
Thumb+Index+Middle+Ring	Little	2.70%
Middle+Ring+Little	Thumb+Index	4.05%
Thumb+Ring+Little	Index+Middle	2.03%
Thumb+Index+Little	Middle+Ring	3.38%
Thumb+Index+Middle	Ring+Little	4.73%
Ring+Little	Thumb+Index+Middle	4.73%
Thumb+Little	Index+Middle+Ring	6.08%
Thumb+Index	Middle+Ring+Little	8.11%
MSALBP(P=8,R=2) of CASIAMS (S460) database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	2%
Thumb+Middle+Ring+Little	Index	5%
Thumb+Index+Ring+Little	Middle	5%
Thumb+Index+Middle+Little	Ring	3%
Thumb+Index+Middle+Ring	Little	6%
Middle+Ring+Little	Thumb+Index	8%
Thumb+Ring+Little	Index+Middle	11%
Thumb+Index+Little	Middle+Ring	6%
Thumb+Index+Middle	Ring+Little	7%
Ring+Little	Thumb+Index+Middle	15%
Thumb+Little	Index+Middle+Ring	19%
Thumb+Index	Middle+Ring+Little	11%

Table 6.7: Removing a finger or fingers from the FCFNN using the ELLBP feature extraction

ELLBP(N=17) of PolyU3D2D database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	0.34%
Thumb+Middle+Ring+Little	Index	0.45%
Thumb+Index+Ring+Little	Middle	0.34%
Thumb+Index+Middle+Little	Ring	0.23%
Thumb+Index+Middle+Ring	Little	0.45%
Middle+Ring+Little	Thumb+Index	1.02%
Thumb+Ring+Little	Index+Middle	0.79%
Thumb+Index+Little	Middle+Ring	0.79%
Thumb+Index+Middle	Ring+Little	1.02%
Ring+little	Thumb+Index+Middle	1.58%
Thumb+Little	Index+Middle+Ring	1.47%
Thumb+Index	Middle+Ring+Little	1.69%
ELLBP(N=17) of IIT Delhi database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	2.03%
Thumb+Middle+Ring+Little	Index	1.35%
Thumb+Index+Ring+Little	Middle	2.03%
Thumb+Index+Middle+Little	Ring	2.70%
Thumb+Index+Middle+Ring	Little	4.05%
Middle+Ring+Little	Thumb+Index	2.03%
Thumb+Ring+Little	Index+Middle	2.03%
Thumb+Index+Little	Middle+Ring	4.05%
Thumb+Index+Middle	Ring+Little	6.08%
Ring+Little	Thumb+Index+Middle	2.70%
Thumb+Little	Index+Middle+Ring	7.43%
Thumb+Index	Middle+Ring+Little	8.11%
ELLBP(N=7) of CASIAMS (S460) database		
Fingers fusion	**Missing finger**	**EER**
Index+Middle+Ring+Little	Thumb	2%
Thumb+Middle+Ring+Little	Index	2%
Thumb+Index+Ring+Little	Middle	1%
Thumb+Index+Middle+Little	Ring	4%
Thumb+Index+Middle+Ring	Little	3%
Middle+Ring+Little	Thumb+Index	3%
Thumb+Ring+Little	Index+Middle	5%
Thumb+Index+Little	Middle+Ring	4%
Thumb+Index+Middle	Ring+Little	5%
Ring+Little	Thumb+Index+Middle	11%
Thumb+Little	Index+Middle+Ring	11%
Thumb+Index	Middle+Ring+Little	10%

one of the important advantages of this network. This has been investigated in Tables 6.6 and 6.7, where they indicate the flexibility of the proposed FCFNN method as any finger connections can be simply removed from the FCFNN structure. Whereas, in the FLFPNN it is not feasible to remove connections between the input layer and the pattern layer because of the missing FT elements, then, changing the full structure of the PNN.

The reason for choosing the neighbouring fingers in Tables 6.6 and 6.7, such as ring+little and middle+ring+little, is that the neighbouring fingers are more likely to be accidentally amputated in reality than separate fingers. These tables firstly show that each finger has a contribution, for example in the MSALBPMS (S460) database in Table 6.6 the remaining fingers after removing the thumb obtained EER value equal to 2%, after ignoring the index or middle finger the EER value achieved 5%, after deleting the ring finger 3% was reported for the EER and after neglecting the little finger the EER value was attained 6%. Furthermore, increasing the number of removed fingers generally reduces the verification performance. For example, in Table 6.7 removing just the thumb from the PolyU3D2D database has affected the verification error rate to be increased and reported 0.34%, continuing removing the index finger with the thumb together have recorded 1.02% and increasing the missing fingers by removing the middle finger in addition to the index and thumb has increased the EER value to be 1.58%. The affected verification performances of the missing finger(s) are different between the feature extraction methods. Therefore, the EER values which have been attained in Table 6.6 are different from those which have been stated in Table 6.7 such as missing the index finger from the CASIAMS (S460) database where it has obtained 5% by using the MSALBP feature extraction, while it has attained 2% by utilizing the ELLBP feature extraction. Also, the EER values vary between the employed databases for the same finger, where each finger has different features or different contribution scores according to the resolution and specifications of its image. Therefore, the little finger in Table 6.6 for instance has recorded 2.70% and 6% for the IIT Delhi and CASIAMS (S460) databases respectively.

Overall evaluating the FLFPNN in terms of missing FT features, the following points are worth highlighting:

- The missing features can be represented by zero.
- The results of the missing and salvaging FT elements are affected by the concatenated feature values.
- The results vary between the employed feature extraction methods.
- The results vary between the databases because of their specifications of resolutions and capturing environments.
- The EER values are dramatically increased when the area of the missing FTs is increased before using the salvage process.
- The improvement in the EER percentages are significantly enhanced after salvaging the missing data, when the area of the missing FTs is enlarged.

On the other hand, the following points can be highlighted in the case of missing finger(s) from the FCFNN architecture:

- It is quite easy to remove the missing finger from the network structure or just ignore its calculations.
- It is also easy to remove more than one finger from the FCFNN computations.
- The contribution of each finger is clearer and the results are dependent on the score contribution fusion of the used fingers in the summation layer.
- The verification performance is varying according to the employed feature extraction method.
- The verification performance is varying according to the specifications of the applied database.
- The verification performance is commonly decreased if the number of missing fingers is increased.

Finally, it can be recommended that using the FCFNN with the ELLBP feature extraction is appropriate for human verification based on the FT. If any amputation happened to a phalanx, two phalanxes or a single finger the salvage approach can be used to reduce the risk of obtaining a wrong verification in the FLFPNN. The flexibility of the FCFNN in its architecture has advantage of

obtaining the verification decision based on the score contributions of the remaining fingers.

6.6 Summary

In this chapter, an efficient human verification system is designed and explained based on the fusion between the feature extractions of the FTs of the five fingers with the PNN called the FLFPNN. Furthermore, an innovative fusion method called the FCFNN has been proposed and implemented. Both fusion methods can be used as multi-object verifications. The key idea of the FCFNN method is to use the fingers' contribution scores in the verification decision as each finger has a different contribution level to provide the personal verification. This network has an advantage in its flexible structure. So, if any finger is accidentally amputated, it can be easily ignored by removing its connections. In this case, the final verification decision will depend on the contribution of the remaining fingers.

Moreover, the FT verification performance was examined for both fusion methods under the possibility of amputating one phalanx, two phalanxes or full finger. This can be considered as a prior investigation in this biometric type. Consequently, a proposed approach has been suggested to solve the problems of removing elements from the FT by salvaging the missing data. The proposed salvage approach was confirmed to successively enhance the verification performance percentages especially with the FLFPNN. On the other hand, removing one or more fingers from the FCFNN was implemented because of its adaptable architecture in this case.

The conclusions of this thesis will be given in the next chapter, by highlighting the contributions of this thesis and declaring the main findings. Future work will also be suggested.

Chapter 7

Conclusions and Future Work

7.1 Introduction

This study covered the area of employing signal processing and machine learning techniques to perform human verification with Finger Textures (FTs). This chapter is divided into two main categories:

- A brief conclusion of this thesis is provided, and
- Some recommendations for future work will be given.

First of all, a brief and full conclusion will be presented for each part of the thesis contributions. That is, the specifications of the Robust Finger Segmentation (RFS) and the Adaptive and Robust Finger Segmentation (ARFS) methods to isolate the finger images will be stated. The main findings of applying the proposed Multi-scale Sobel Angles Local Binary Pattern (MSALBP) and Enhanced Local Line Binary Patterns (ELLBP) will be explained in terms of feature extractions. The differences between employing the finger fusion methods will be illustrated for the Feature Level Fusion with the Probabilistic Neural Network (FLFPNN) and the Finger Contribution Fusion Neural Network (FCFNN). The main findings of examining the missing FT parts will be concluded and the benefits of applying the suggested salvaging approach will be described. Moreover, the processing steps of the novel approach of producing the Receiver Operating Characteristic (ROC) graph from the Probabilistic Neural Network (PNN) and from the FCFNN will be covered.

7.2 Conclusions and Main Findings

7.2.1 Descriptions of Employed Databases

Three databases with a large number of contactless finger samples were employed in this thesis. The first database was from the Hong Kong Polytechnic University Contact-free 3D/2D (PolyU3D2D) Hand Images Database (Version 1.0) [67]. It consists of 8,850 fingers from 1770 very low resolution hand images. The second database was from the Indian Institute of Technology (IIT) Delhi Palmprint Image Database (Version 1.0) [65,66]. A total of 4,440 finger images were used in this study, where they were collected from 888 high resolution hand images. The participants of the IIT Delhi database were asked to provide high degrees of hand movements during the capturing step. The third database was for a specific spectral wavelength of 460nm, which can be found in the CASIA Multi-Spectral (CASIAMS) Palmprint image database (Version 1.0) [9]. This data has 3,000 fingers included in 600 low resolution hand images. It is valuable to work with this type of database as its FTs have different features than the first and second databases, because they were collected under a single spectrum component.

7.2.2 Finger Segmentation Approaches

Two segmentation approaches were suggested to isolate finger areas from contact-free hand images. Firstly, the proposed RFS method has been introduced to extract the finger images. Several image processing steps were designed to maintain the hand area and detect the fingers. Briefly, these steps are converting the colour image (if there is any) to a grayscale image; applying a binarization operation; removing all the small noises around the largest hand region; complementing the resulting image; deleting the noises which might exist within the hand border; and detecting the finger objects. Henceforth, the main finger points (tip, valley and symmetric) can be determined and used to segment the finger images. It is worth mentioning that additional pixels have been included with each segmented finger image to cover the patterns of the lower knuckles as these patterns can enhance the verification performance according to [25].

Secondly an efficient finger segmentation approach named ARFS was proposed. This approach was inspired from the previous segmentation method. A scanning

line and adaptive rotation were exploited to improve performance. The ARFS can deal with different hand alignment positions such as translation and rotation. It also discovers the fingers by the concept of objects after removing the expected and unexpected noise. Again, the lower knuckles are proposed to be involved in the segmented finger images.

Consequently, segmenting fingers are provided by extracting Region of Interests (ROIs), from which FT parts can be collected. The largest inner rectangle technique [58] is used in extracting the ROI of each FT from a finger image.

The results show that hand finger images are successfully isolated and they involve all the captured FT features. This has advantages over prior work, which usually fails to include the full patterns of the lower knuckles. Both RFS and ARFS can maintain the finger images by applying the adaptive threshold strategy. Furthermore, the ARFS can address various hand alignment movements (translation, scaling and rotation), because of the facilities of the adaptive rotation and scanning line strategies.

The main disadvantage of both segmentation approaches is that they require many image processing steps and this can lead to time consuming.

7.2.3 Feature Extraction Approaches

Two essential feature extraction contributions were proposed in this thesis. The first major contribution was introducing a new feature extraction method called the MSALBP. This feature extraction method fuses between the Sobel orientation angles, which are calculated from the horizontal and vertical Sobel attributes of the FT, and the Multi-Scale Local Binary Pattern (MSLBP). The resulting combinations are blocked into non-overlapping blocks and statistical calculations are implemented to form an input vector for an Artificial Neural Network (ANN).

The second contribution is the ELLBP. This is an enhanced version of the Local Line Binary Pattern (LLBP). Due to the fact that the FT generally involves two main patterns (vertical and horizontal), two analysed vectors from the LLBP are required; a horizontal vector is therefore assigned for the vertical patterns and a vertical vector is assigned for the horizontal patters. In the ELLBP effective texture values have been calculated by applying the weighted summation fusion rule between the computed horizontal and vertical codes. Likewise for the

MSALBP, similar blocking operations and statistical calculations are used to collect the extracted features.

Extensive experiments were performed to determine the best parameters for both feature extraction methods. Significant improvements were observed in the verification performance after applying both suggested approaches compared to various updated Local Binary Pattern (LBP) versions. The best result for the PolyU3D2D database was with the ELLBP, where the EER was equal to 0.34% reflecting an improvement of 50% from the nearest compared EER value. Whilst, the best results attained for the IIT Delhi database were recorded for both MSALBP and ELLBP, where they achieved the lowest EER value of 1.35% resulting in improvement of 33.50% from the closest compared EER value. For the CASIAMS (S460), the best EER value achieved was 2% for the MSALBP, where the improvement was 60% from the nearest compared EER, and 0% for the ELLBP, where this was the best obtained result. In general, the ELLBP has recorded better results than the MSALBP. Nevertheless, the MSALBP operator achieved the fastest execution time compared with all of the updated LBP versions including the ELLBP.

It can be concluded that the suggested MSALBP and ELLBP approaches showed remarkable performance in the case of personal FT verifications. On the other hand, it has been investigated that the MSALBP is not robust against the illumination compared with the ELLBP and the ELLBP needs additional time to obtain the values of the weights in its weighted summation equation.

7.2.4 Finger Fusions

Two essential fusion methods were presented in this thesis. Firstly, a powerful verification structure was described and implemented by using the FLFPNN. This fusion method employs a feature level fusion technique, where the extracted feature vectors of the fingers were combined based on the concatenation rule. Then, a PNN is applied to assess the FT verification performance.

Secondly, a novel FCFNN method was introduced. The key idea of this fusion is to consider the contribution score of each finger in the verification decision, due to the fact that each finger has a contribution score different from others. Thus, an innovation score fusion was described by creating a new hidden layer, named the

contribution layer, inside the FCFNN.

In fact, the previous presented results in Subsection 7.2.3 were recorded according to the FLFPNN, where it was considered as a standard fusion method. Consequently, such interesting results were observed after using the suggested FCFNN method. To illustrate, the verification error rates for the PolyU3D2D database have been attained 0.68% by using the MSALBP with the FLFPNN and 0.34% by utilizing the ELLBP with the FLFPNN. These percentages have been noticeably reduced to 0.23% after utilizing the MSALBP with the FCFNN and 0.11% after using the ELLBP with the FCFNN. For the IIT Delhi database, the EER has been slightly increased from 1.35% to 0.23% by employing the MSALBP with the FLFPNN and the MSALBP with the FCFNN, respectively. Whereas, the EER has obtained the same value of 1.35% by applying the ELLBP to either the FLFPNN or the FCFNN. This can be explained by understanding the specifications of the IIT Delhi database, where different hand positions exist. So, slightly dislocated FTs are likely to be collected and could cause a wrong verification decision as it has been reported with the MSALBP, whilst, the ELLBP could maintain the EER value because it is more robust than the MSALBP. For the CASIAMS (S460) database the EER values showed stable performance in both fusion methods, where it was 2% for the MSALBP and 0% for the ELLBP.

As a general conclusion regarding both fusion methods, the FCFNN can achieve better performance than the FLFPNN because it utilizes the score contribution levels of fingers. Nevertheless, the FCFNN has a sensitive reaction toward the variation values in its inputs, but it appears that there is no serious drawback of employing this type of fusion as the result may only be slightly decreased. Finally, the FCFNN may maintain the verification performance when the best results are obtained.

7.2.5 Missing FT features

The experiments have been extended to examine the verification performance under the condition of missing FT elements. For example, removing a distal phalanx, distal and intermediate phalanxes or even a full finger. Zero values were employed to cover the missing parts of a finger image. Both FLFPNN and FCFNN have been evaluated and analysed in this case.

A novel approach is also suggested and implemented to salvage the missing elements by taking advantage from the embedded weights in the trained PNN. Briefly, each node in the hidden layer in the PNN already saves the full training pattern during the enrolment phase. So, the nearest training pattern to the provided FTs can be used to form the missing features.

Experimental results show that the verification performance could be affected by missing a part of a finger depending on the size of the missing area, the applied feature extraction method and the employed finger fusion method. For instance, the achieved EER values before and after missing FT elements by applying the ELLBP were in general smaller than the obtained EER values by using the MSALBP in the same cases. Also in the FLFPNN, increasing the size of the missing features could dramatically influence the verification performance. Whilst, increasing the size of the missing FTs could produce a stable verification effect by using the FCFNN, because of the supported score contributions of other fingers.

The proposed salvaging method confirms its ability to enhance the verification performance, where remarkable improvements could be seen by observing the EER values after the FLFPNN. These improvements were increased with more missing FT elements. However, this was not the same case in the FCFNN, where the improvements of the verification performance were reduced after increasing the missing features. This is because the FCFNN is influenced by the small missing FT areas, as mentioned.

On the other hand, the FCFNN has been found to be more flexible than the FLFPNN in removing a single finger or more, due to its applicable architecture. To illustrate, eliminating determined connections, which were assigned for a finger or fingers, can be easily implemented in this network. Additional experiments have been applied in this matter. So, the FCFNN has been tested under the suggestions of ignoring a single finger or multiple nearest fingers, due to the fact that the nearest fingers are usually being under the danger of amputation compared with separated fingers. The results revealed that deleting more fingers negatively influenced the EER values. Furthermore, it has been noticed that the verification performance were affected by applying the feature extraction method and the contribution score of each finger.

Finally, it can be concluded that employing the salvage approach can be so effective when using the FLFPNN method, whereas, the FCFNN is efficient

enough to be utilized in terms of FT verifications.

7.2.6 ROC graphs

A novel method to generate an ROC graph from a PNN was suggested. The main idea of this issue is to extract the score values from each single class of the PNN. These values can be found in the summation layer of this network, where the summation nodes hold the actual output values. Furthermore, these values were remapped or recalculated according to their relationships with the class targets, where the relationship between the summation layer values and the target class has been established and implemented. This method seems more efficient than using the logical output values from the decision layer. After collecting the ROC parameters (the False Acceptance Rate (FAR) and the False Rejection Rate (FRR)) from each single class, a relationship could be constructed to produce the ROC curve. Afterwards, each ROC parameter was combined to all single classes by applying the average operation. Hence, a smooth ROC curve has been produced to represent the performance of the multiple classes. Different methods have been used to assess the work. Generating the ROC graph has never been considered in the multi-class PNN and a substantial literature review was undertaken to confirm this fact.

Similar strategies were applied to establish the ROC graph from the FCFNN. Again, The main processes of generating the ROC curve are: collecting the effective FCFNN output values from the summation layer; remapping these values according to the FCFNN classifications; computing the FAR and the FRR for each class; generating one FAR and FRR for all classes by calculating the averages of FARs and FRRs respectively; producing a relationship between the computed FAR and (1-FRR); and finally depicting the ROC curve.

Each of the proposed approaches of generating the ROC curves has a disadvantage, which can be found in remapping the extracted score values as the relationship between these values and the class targets of the employed classifier is not a straightforward task.

7.3 Future Work

The findings of this work have a number of important implications for future studies. So, some of these findings can provide the following insights for future research:

- The databases which have been used in this thesis are fundamentally established for palm print studies. So, it is believed that a specific FT database is required, where this database has to cover the full FT region and involve all of its feature types (wrinkles, visible lines and ridges).
- In this thesis, only two types of FT features have been used which are the wrinkles and knuckle lines as just these types are provided. A future study can investigate the ridge patterns of the FT.
- In the case of finger segmentations, additional efforts will be required to generate a robust ground truth. The current suggested ground truth is based on the essential points of fingers (tips, valleys and symmetric). Obviously, these points can not cover all the patterns of the lower knuckles. So, a justified ground truth for the finger segmentation can be established then provided to other researchers.
- Additional feature extraction methods can be suggested to employ adaptive horizontal and vertical weights in the ELLBP. A ratio between the horizontal and vertical codes may provide a good basis to calculate the adaptive weights.
- A deep learning technique can be employed to extract the FT features and verify the users, provided sufficient data are available.
- In this study, the main classifier or matcher was the PNN, then this network has been developed to the FCFNN. In future investigations, it is possible to use a different matcher such as the Support Vector Machine (SVM).
- The proposed FCFNN can be efficiently utilized with other biometric characteristics. For instance, the feature extractions from the vein patterns of the four parts of the sclera, where four feature vectors can be collected from the white region on the right and left side of each eye. Thus, four feature vectors will be created. From this point, the FCFNN can be fed with these vectors in the input layer and four contribution score vectors will be calculated then considered in the decision layer.

- Building upon the previous point, the FCFNN can be used to compute the contribution scores of each knuckle in a finger knuckle biometric system. In this case, a biometric system based on the outer knuckles or the inner knuckles can be created. It is expected that some knuckles will have higher contribution scores than the others. Examples of these are the middle outer knuckles which may produce contribution scores higher than the upper outer knuckles and the upper inner knuckles may generate contribution scores less than the lower inner knuckles.

- The FCFNN is easily used in multi-modal biometric systems where a fusion between two or more biometric characteristics can be implemented by this network. It can be observed that some biometric characteristics have more influenced contribution than the others on a multi-modal recognition system. For example, the Finger Veins (FV) can have a higher recognition contribution than the Finger Geometry (FG).

- It is noteworthy that the salvaging approach could be exploited in many biometric applications such as the iris print, where a part of the iris texture near the pupil may be accidentally removed after an eye lens operation.

- After generating the ROC curve, the Detection Error Tradeoff (DET) graph is easy to be constructed from the relationships between the FAR and FRR. Moreover, a suitable determined value can be chosen for the FAR and its corresponding value from the FRR in order to increase or decrease the security level of the system.

- A robust biometric system can be established by employing the inner and outer finger surfaces at the same time, where the outer knuckles can be used with the FTs in that system. Furthermore, another biometric system can be proposed to capture the FG of fingers with the FT.

- This thesis is assigned for human verification. Additional studies are required for personal identifications and classifications.

Appendix A

Receiver Operating Characteristic

A.1 Introduction

Traditionally, an Receiver Operating Characteristic (ROC) curve has been used widely to report the recognition system measurements. It has been remarked that generating an ROC graph for a Probabilistic Neural Network (PNN) is not a straightforward task. Similarly, constructing an ROC graph for the Finger Contribution Fusion Neural Network (FCFNN) is not provided. Therefore, two contributions are considered:

- A new approach to generate the ROC graph from a multi-class PNN has been presented.
- A modified method to produce the ROC curve from the proposed FCFNN has been illustrated.

An ROC graph is a measuring method used widely to evaluate verification or identification systems. It consists of different parameters, the False Acceptance Rate (FAR), False Rejection Rate (FRR) and True Positive Rate (TPR) [130]. Furthermore, the trade off point which is considered as an essential parameter to evaluate any recognition system is the Equal Error Rate (EER). Basically, if the EER has a small value this means that the system is efficient and if the EER has a large value this reports that the recognition is inefficient.

The rest of this appendix is organized as follows. Section 2 presents the theoretical concepts of the ROC, Section 3 illustrates how to produce the ROC from the PNN. Section 4 describes generating the ROC curve from the FCFNN structure. Section 5 provides the summary.

A.2 Theoretical Concept of the ROC

Generating the ROC graph has never been considered in the multi-class PNN before the work of Al-Nima *et al.* [27] and a substantial literature review was undertaken to confirm this fact. Table A.2 provides a summary of related publications reported in the literature, which have been covered in Chapter 2:Section 2.3.4.
The basic ROC parameters can be computed according to the following outcomes:

- True Positive (TP), if the neural output has correctly classified the positive case.
- False Positive (FP), if the neural output has incorrectly classified the positive case.
- True Negative (TN), if the neural output has correctly classified the negative case.
- False Positive (FN), if the neural output has incorrectly classified the negative case.

Thus, a confusion matrix is found as shown in Table A.1.

Table A.1: The confusion matrix

TP	FP
FN	TN

Principally, according to an adaptive threshold the percentage of the correctly classified genuines in a recognition process will be considered as TPR and the percentage of the incorrectly classified impostors will be recorded as FAR [130]. With more explanations, a relationship will be established between the FAR and TPR (mathematically this equals to 1-FRR) for each classifier [130] at each threshold. FAR represents a matching value which is greater than the threshold and FRR represents a matching value which is less than the threshold [30].
So, the following equations can be computed and collected for each thresholding

Table A.2: Summary of related approaches for generating the ROC graph from a PNN

Reference	**Existing ROC curve?**	**Problem of ROC curve**
Woods and Bowyer [130]	Yes	Utilizing the bias of the hidden layer. It is just for a MLP with a bias and two classes only
Orr [131]	Yes	Using commercial software for PNN with two classes only. The curve was not smooth
Ooi *et al.* [132]	Yes	The results were averaged after running the PNN many times
Sharma *et al.* [133]	Yes	A combination method using a Parzen with the PNN
Joshi *et al.* [135]	Yes	A combination method using a Parzen with the PNN
Almaadeed *et al.* [136]	Yes	A Combination of multiple neural networks then applying a voting process
Lin *et al.* [137]	No	—
Huang *et al.* [138]	No	—
Meshoul and Batouche [139]	No	—
Hossain and Amin [140]	No	—
Sundaram and Dhara [141]	No	—
Chuang *et al.* [142]	No	—
Sudha and Bhavani [143]	No	—
Yu and Huang [144]	No	—
El-Alfy [40]	No	—
Saini *et al.* [145]	No	—
Wibawa *et al.* [146]	No	—
Perwira *et al.* [147]	No	—

instance [121]:

$$FRR = \frac{Number\ of\ Rejected\ Clients}{Total\ Number\ of\ Clients} \times 100\% \tag{A.1}$$

$$FAR = \frac{Number\ of\ Accepted\ Imposters}{Total\ Number\ of\ Imposters} \times 100\% \tag{A.2}$$

It has been illustrated that reducing the FAR value will increase the security level of the system, whilst, increasing the FAR value will increase the flexibility of the system in terms of accepting input subjects [182].

A.3 Generation of ROC for PNN

Two main problems arise when using the PNN. Firstly, the PNN outputs are always logical (zeros and one); secondly, a PNN is considered as a multi-class classifier, because it usually has more than one output class. To solve these problems a new approach to acquire the score values from the PNN, establish the relationship between the ROC parameters for each class and fusing them to generate one main ROC curve has been proposed.

Generally, ROC parameters are calculated for each single class of a neural network. Assuming an Multi-Layer Perceptron (MLP) network has one output node to classify two classes, O represents the neural network output values and T represents the target with one and zero. A set of $\{O,T\}$ will be arranged and an adaptive threshold is applied through this set [130, 183]. In other words, the conventional method for establishing the ROC curve is by varying a threshold through the output node, where the output values are considered as scores and the relationship is constructed with its targets.

The main challenge in producing the ROC curve is how to get the score values, which are one of the main parameters of the $\{O,T\}$ set in this matter. Fundamentally, the summation layer will give the exact probabilistic values of each class for the same input vector. Hence, the decision layer of the PNN picks the maximum of the summation values S_j and provides the target class for the input vector [139], as shown below:

$$Tclass_j = \begin{cases} 1 & \text{if } S_j = max \\ 0 & \text{otherwise} \end{cases}, \quad j = 1, 2, ..., c \tag{A.3}$$

where, $Tclass$ is the desired target.

It could be argued that the actual output is in the summation node of the jth class and the output layer is just a logical decision of the corresponding values in

the summation layer. However, after analysing these values it can be seen that they require a remapping process. That is because, according to Equation (A.3) some very small values could win the competition. So, to address this problem a relationship between the outputs of the summation layer and the target class has been established and implemented according to the following equation:

$$PNN_{Score_j} = \begin{cases} PNN_{Score_j} \times Fac1 & \text{if} \quad Tclass_j = 1 \\ PNN_{Score_j} \times Fac2 & \text{if} \quad Tclass_j = 0 \end{cases} \tag{A.4}$$

where, PNN_{Score} is the output of the summation nodes, $Fac1$ and $Fac2$ are scaling factors and $Fac2$ can be denoted as $Fac2 = 1/Fac1$.
After collecting the ROC parameters (TPR and FAR) it is easy to draw the ROC curve for each class as there is commonly a relationship between them. To combine this process for all classes in a multi-class PNN, average processes can be performed for all TPRs and FARs. This will lead to the generation of the main smoothing ROC curve that describes the PNN performance.

A.4 Generation of ROC for FCFNN

The aim of this section is to present a method to produce an ROC curve from an FCFNN. It has been found that the same method of producing the ROC curve which has been used in the PNN can be applied in this method. Basically, Equation (A.3) can be utilized in the FCFNN too, however, the summation layer in this network is collecting its values from the contribution layer as explained in Chapter 6:Subsection 6.3.2. Moreover, Equation (A.5) can be updated to:

$$FCFNN_{Score_j} = \begin{cases} FCFNN_{Score_j} \times Fac1 & \text{if} \quad Tclass_j = 1 \\ FCFNN_{Score_j} \times Fac2 & \text{if} \quad Tclass_j = 0 \end{cases} \tag{A.5}$$

where $FCFNN_{Score}$ is the output scores of the summation layer.
All operations which have been reported in the PNN to get the ROC graph for each class are utilized in the FCFNN. Subsequently, two average processes have been implemented for all classes. One for the FAR values and another one for the TPR values. This will combine the ROC parameters for all output classes of the FCFNN

together and produce one ROC curve referring to the total performance.

A.5 Conclusion of Appendix A

The key idea of this study was to extract the score values from each single class of the PNN and FCFNN. These values can be found in the summation layer of these networks, where the summation nodes hold the actual output values. However, these values were remapped or recalculated according to their relationships with the class targets, where the relationship between the summation layer values and the target class has been established and implemented. This method seems more efficient than using the logical output values from the decision layer. After collecting the ROC parameters (FAR and TPR) from each single class, a relationship could be constructed to produce the ROC curve. Each ROC parameter was fused or combined to all single classes by using the average operation. Afterwards, a beneficial demonstrated ROC curve has been produced to represent the multiple classes by using large databases and different methods in this thesis, which confirm the value of this work.

References

[1] "Ayat," http://quran.ksu.edu.sa/index.php?l=ar#aya=46_15&m=hafs&qaree=husary&trans=en_sh, Accessed: 2017-05-02.

[2] "Fvc2000: Fingerprint Verification Competition," http://bias.csr.unibo.it/fvc2000/download.asp, accessed: 2017-01-18.

[3] S. Ribaric and I. Fratric, "An online biometric authentication system based on eigenfingers and finger-geometry," in *13th European Signal Processing Conference*, 2005, pp. 1–4.

[4] Z. Liu and S. Song, "An embedded real-time finger-vein recognition system for mobile devices," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 522–527, 2012.

[5] M. Amraoui, M. El Aroussi, R. Saadane, and M. Wahbi, "Finger-knuckle-print recognition based on local and global feature sets," *Journal of Theoretical & Applied Information Technology*, vol. 46, no. 1, pp. 54–60, 2012.

[6] "IIT Delhi Finger Knuckle Database version 1.0." [Online]. Available: http://www.comp.polyu.edu.hk/~csajaykr/knuckle/iitd_knuckle.htm

[7] M. Liu, Y. Tian, and L. Lihua, "A new approach for inner-knuckle-print recognition," *Journal of Visual Languages & Computing*, vol. 25, no. 1, pp. 33–42, 2014.

[8] Q. Li, Z. Qiu, D. Sun, and J. Wu, "Personal identification using knuckleprint," in *Advances in Biometric Person Authentication*. Springer, 2004, pp. 680–689.

[9] "CASIA-MS-PalmprintV1." [Online]. Available: http://biometrics.idealtest.org/

[10] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2228–2244, 2012.

[11] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," in *3rd International Conference on Image and Graphics (ICIG)*, 2004, pp. 306–309.

[12] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.

[13] Y. Tong, R. Chen, and Y. Cheng, "Facial expression recognition algorithm using LGC based on horizontal and diagonal prior principle," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 16, pp. 4186–4189, 2014.

[14] A. Goda, "Introduction to neural networks," http://home.agh.edu.pl/~vlsi/AI/intro/, 2005, accessed: 2016-10-23.

[15] M. A. M. Abdullah, S. S. Dlay, W. L. Woo, and J. A. Chambers, "Robust iris segmentation method based on a new active contour force with a noncircular normalization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[16] M. Frucci, M. Nappi, D. Riccio, and G. S. di Baja, "Wire: Watershed based iris recognition," *Pattern Recognition*, vol. 52, pp. 148–159, 2016.

[17] Z. Zhou, E. Du, N. Thomas, and E. Delp, "A new human identification method: Sclera recognition," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 3, pp. 571–583, 2012.

[18] S. Alkassar, W. Woo, S. Dlay, and J. Chambers, "Robust sclera recognition system with novel sclera segmentation and validation techniques," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–13, 2016.

[19] M. De Marsico, M. Nappi, D. Riccio, and H. Wechsler, "Robust face recognition for uncontrolled pose and illumination changes," *IEEE*

Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 1, pp. 149–163, 2013.

[20] A. A. M. Al-Shiha, W. L. Woo, and S. S. Dlay, "Multi-linear neighborhood preserving projection for face recognition," *Pattern Recognition*, vol. 47, no. 2, pp. 544–555, 2014.

[21] F. Yue, B. Li, M. Yu, and J. Wang, "Hashing based fast palmprint identification for large-scale databases," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 769–778, 2013.

[22] R. Raghavendra and C. Busch, "Novel image fusion scheme based on dependency measure for robust multispectral palmprint recognition," *Pattern recognition*, vol. 47, no. 6, pp. 2205–2221, 2014.

[23] M. T. S. Al-Kaltakchi, W. L. Woo, S. S. Dlay, and J. A. Chambers, "Study of statistical robust closed set speaker identification with feature and score-based fusion," in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2016, pp. 1–5.

[24] ——, "Study of fusion strategies and exploiting the combination of MFCC and PNCC features for robust biometric speaker identification," in *4th International Conference on Biometrics and Forensics (IWBF)*, March 2016, pp. 1–6.

[25] R. R. Al-Nima, S. S. Dlay, W. L. Woo, and J. A. Chambers, "Human authentication with finger textures based on image feature enhancement," in *2nd IET International Conference on Intelligent Signal Processing (ISP)*, 2015.

[26] R. R. O. Al-Nima, S. S. Dlay, S. A. M. Al-Sumaidaee, W. L. Woo, and J. A. Chambers, "Robust feature extraction and salvage schemes for finger texture based biometrics," *IET Biometrics*, vol. 6, no. 2, pp. 43–52, 2016.

[27] R. R. O. Al-Nima, S. S. Dlay, W. L. Woo, and J. A. Chambers, "A novel biometric approach to generate ROC curve from the probabilistic neural network," in *24th IEEE Signal Processing and Communication Application Conference (SIU)*, 2016, pp. 141–144.

[28] P. Yu, D. Xu, and H. Zhou, "Feature level fusion using palmprint and finger geometry based on canonical correlation analysis," in *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, 2010, pp. V5–260–V5–264.

[29] M. Ferrer, A. Morales, C. Travieso, and J. Alonso, "Low cost multimodal biometric identification system based on hand geometry, palm and finger print texture," in *41st Annual IEEE International Carnahan Conference on Security Technology*, 2007, pp. 52–58.

[30] Ş. Sağiroğlu and N. Özkaya, "An intelligent face features generation system from fingerprints," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 17, no. 2, pp. 183–203, 2009.

[31] R. R. Al-Nima, S. Dlay, and W. Woo, "A new approach to predicting physical biometrics from behavioural biometrics," *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, no. 11, pp. 1996–2001, 2014.

[32] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[33] V. Matyáš and Z. Rıha, "Biometric authentication systems," in *verfügbar über: http://grover. informatik. uni-augsburg. de/lit/MM-Seminar/Privacy/riha00biometric. pdf.* Citeseer, 2000.

[34] W. Woo and S. Dlay, "A novel fingerprint pressure deformation algorithm for biometric authentication," *WSEAS Transactions on Communications*, vol. 4, no. 12, pp. 1373–1380, 2005.

[35] S. Modi, S. Elliott, J. Whetsone, and H. Kim, "Impact of age groups on fingerprint recognition performance," in *IEEE Workshop on Automatic Identification Advanced Technologies*, 2007, pp. 19–23.

[36] R. Donida Labati, A. Genovese, V. Piuri, and F. Scotti, "Toward unconstrained fingerprint recognition: A fully touchless 3-d system based

on two views on the move," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 2, pp. 202–219, 2016.

[37] M. A. Abdullah, J. A. Chambers, W. L. Woo, and S. S. Dlay, "Iris biometric: Is the near-infrared spectrum always the best?" in *3rd Asian Conference on Pattern Recognition (ACPR2015)*, 2015.

[38] M. Abdullah, F. Al-Dulaimi, W. Al-Nuaimy, and A. Al-Ataby, "Efficient small template iris recognition system using wavelet transform," *International Journal of Biometric and Bioinformatics*, vol. 5, no. 1, pp. 16–27, 2011.

[39] P. Varchol and D. Levicky, "Using of hand geometry in biometric security systems," *Radioengineering-Prague-*, vol. 16, no. 4, p. 82, 2007.

[40] E. S. M. El-Alfy, "Automatic identification based on hand geometry and probabilistic neural networks," in *5th International Conference on New Technologies, Mobility and Security (NTMS)*, May 2012, pp. 1–5.

[41] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[42] S. A. Al-Sumaidaee, S. S. Dlay, W. L. Woo, and J. A. Chambers, "Facial expression recognition using local gabor gradient code-horizontal diagonal descriptor," in *2nd IET International Conference on Intelligent Signal Processing (ISP)*, 2015.

[43] S. V. Tathe, A. S. Narote, and S. P. Narote, "Face detection and recognition in videos," in *IEEE Annual India Conference (INDICON)*, Dec 2016, pp. 1–6.

[44] G. Michael, T. Connie, and A. Jin, "Robust palm print and knuckle print recognition system using a contactless approach," in *5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2010, pp. 323–329.

[45] G. K. O. Michael, T. Connie, and A. T. B. Jin, "An innovative contactless palm print and knuckle print recognition system," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1708–1719, 2010.

[46] R. Raid, "Human authentication with earprint for secure telephone system," *Iraqi Journal of Computers,Communications,Control and Systems Engineering*, vol. 12, no. 2, pp. 47–56, 2012.

[47] M. Oravec, J. Pavloviov, D. Sopiak, V. Jirka, M. Loderer, . Lehota, M. Vodika, M. Fakovec, M. Mihalik, M. Tomk, and J. Gert, "Mobile ear recognition application," in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2016, pp. 1–4.

[48] A. Abaza, A. Ross, C. Hebert, M. A. F. Harrison, and M. S. Nixon, "A survey on ear biometrics," *ACM computing surveys (CSUR)*, vol. 45, no. 2, p. 22, 2013.

[49] T. R. Borah, K. K. Sarma, and P. H. Talukdar, "Retina recognition system using adaptive neuro fuzzy inference system," in *International Conference on Computer, Communication and Control (IC4)*, 2015, pp. 1–6.

[50] I. V. Anikin and E. S. Anisimova, "Handwritten signature recognition method based on fuzzy logic," in *Dynamics of Systems, Mechanisms and Machines (Dynamics)*, Nov 2016, pp. 1–5.

[51] D. Morocho, A. Morales, J. Fierrez, and R. Vera-Rodriguez, "Towards human-assisted signature recognition: Improving biometric systems through attribute-based recognition," in *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, Feb 2016, pp. 1–6.

[52] S. Shorrock, A. Yannopoulos, S. Dlay, and D. Atkinson, "Biometric verification of computer users with probabilistic and cascade forward neural networks," *Advances in Physics, Electronics and Signal Processing Applications*, pp. 267–272, 2000.

[53] S. Mondal and P. Bours, "Person identification by keystroke dynamics using pairwise user coupling," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2017.

[54] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[55] Y. Makihara, D. S. Matovski, M. S. Nixon, J. N. Carter, and Y. Yagi, "Gait recognition: Databases, representations, and applications," *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2015.

[56] D. Muramatsu, Y. Makihara, H. Iwama, T. Tanoue, and Y. Yagi, "Gait verification system for supporting criminal investigation," in *2nd IAPR Asian Conference on Pattern Recognition*, Nov 2013, pp. 747–748.

[57] A. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Springer Science+Business Media, LLC, 2008.

[58] V. Kanhangad, A. Kumar, and D. Zhang, "A unified framework for contactless hand verification," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1014–1027, 2011.

[59] M. I. Ahmad, "Feature extraction and information fusion in face and palmprint multimodal biometrics," PhD thesis, School of Electrical and Electronic Engineering, Newcastle University, 2013.

[60] M. Anitha and K. R. Rao, "Fusion of finger inner knuckle print and hand geometry features to enhance the performance of biometric verification system," *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 10, no. 10, pp. 1310–1315, 2016.

[61] ISO and IEC, "Information technology vocabulary part 37: Biometrics," *International Standard ISO/IEC 2382-37*, pp. 1–21, 2012(E).

[62] H. AlMahafzah, M. Imran, and H. Sheshadri, "Multibiometric: Feature level fusion using FKP multi-instance biometric," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 3, 2012.

[63] G. Amayeh, G. Bebis, and M. Hussain, "A comparative study of hand recognition systems," in *International Workshop on Emerging Techniques and Challenges for Hand-Based Biometrics*, Aug 2010, pp. 1–6.

[64] B. Bhaskar and S. Veluchamy, "Hand based multibiometric authentication using local feature extraction," in *International Conference on Recent Trends in Information Technology (ICRTIT)*, 2014, pp. 1–5.

[65] "IIT Delhi Palmprint Image Database version 1.0." [Online]. Available: http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm

[66] A. Kumar, "Incorporating cohort information for reliable palmprint authentication," in *6th IEEE Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP'08.*, 2008, pp. 583–590.

[67] "The Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database version 1.0." [Online]. Available: http://www.comp.polyu.edu.hk/~csajaykr/myhome/database_request/3dhand/Hand3D.htm

[68] U. Halici and G. Ongun, "Fingerprint classification through self-organizing feature maps modified to treat uncertainties," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1497–1512, 1996.

[69] R. Gil, G. D. Orueta, M. Tawfik, F. Garcia-Loro, A. P. Martin, E. Sancristobal, S. Martin, and M. Castro, "Fingerprint verification system in tests in moodle," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 8, no. 1, pp. 23–30, 2013.

[70] E. N. Bifari and L. A. Elrefaei, "Automated fingerprint identification system based on weighted feature points matching algorithm," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2014, pp. 2212–2217.

[71] V. Conti, C. Militello, F. Sorbello, and S. Vitabile, "A frequency-based approach for features fusion in fingerprint and iris multimodal biometric identification systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 384–395, 2010.

[72] R. Mukhaiyar, "Cancellable biometric using matrix approaches," PhD thesis, School of Electrical and Electronic Engineering, Newcastle University, 2015.

[73] A. Makrushin, K. Qian, C. Vielhauer, and T. Scheidat, "Forensic analysis: on the capability of optical sensors to visualize latent fingerprints on rubber gloves," in *3rd IEEE International Workshop on Biometrics and Forensics (IWBF)*, 2015, pp. 1–6.

[74] J. Kotzerke, S. A. Davis, R. Hayes, L. J. Spreeuwers, R. N. J. Veldhuis, and K. J. Horadam, "Discriminating fingermarks with evidential value for forensic comparison," in *3rd IEEE International Workshop on Biometrics and Forensics (IWBF)*, 2015, pp. 1–6.

[75] R. Merkel, J. Dittmann, and M. Hildebrandt, "Latent fingerprint persistence: A new temporal feature space for forensic trace evidence analysis," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 4952–4956.

[76] J. Ezeobiejesi and B. Bhanu, "Latent fingerprint image segmentation using fractal dimension features and weighted extreme learning machine ensemble," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 214–222.

[77] N. Zhang, Y. Zang, X. Yang, X. Jia, and J. Tian, "Adaptive orientation model fitting for latent overlapped fingerprints separation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1547–1556, 2014.

[78] A. A. Paulino, J. Feng, and A. K. Jain, "Latent fingerprint matching using descriptor-based hough transform," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 31–45, 2013.

[79] U. M. Kabadi, "Classification of diabetes according to national diabetic data group," *Diabetes care*, vol. 14, no. 7, pp. 612–613, 1991.

[80] R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos, "Biometric identification through hand geometry measurements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1168–1171, 2000.

[81] O. Ayurzana, B. Pumbuurei, and H. Kim, "A study of hand-geometry recognition system," in *Ifost*, vol. 2, 2013, pp. 132–135.

[82] J. Svoboda, M. M. Bronstein, and M. Drahansky, "Contactless biometric hand geometry recognition using a low-cost 3d camera," in *International Conference on Biometrics (ICB)*, 2015, pp. 452–457.

[83] S. Malassiotis, N. Aifanti, and M. G. Strintzis, "Personal authentication using 3-d finger geometry," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 12–21, 2006.

[84] B. Aghili and H. Sadjedi, "Personal identification/verification by using four fingers," in *3rd International Congress on Image and Signal Processing*, vol. 6, 2010, pp. 2619–2623.

[85] F. Liu, H. Liu, and L. Gao, "Hand recognition based on finger-contour and PSO," in *Proceedings of International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 35–39.

[86] W. Kang and Q. Wu, "Pose-invariant hand shape recognition based on finger geometry," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1510–1521, 2014.

[87] B. J. Kang and K. R. Park, "Multimodal biometric method based on vein and geometry of a single finger," *IET Computer Vision*, vol. 4, no. 3, pp. 209–217, 2010.

[88] A. A. Joshi, P. Deshpande, and A. S. Tavildar, "Enhancing accuracy for personal identification usinghierarchical based fusion of finger geometry and palm print modalities," in *International Conference on Electronics and Communication Systems (ICECS)*, 2014, pp. 1–4.

[89] M. L. Anitha and K. A. R. Rao, "A novel bimodal biometric identification system based on finger geometry and palm print," in *19th International Conference on Digital Signal Processing*, 2014, pp. 574–579.

[90] Y. Lu, S. Yoon, and D. Park, "Finger vein identification system using two cameras," *Electronics Letters*, vol. 50, no. 22, pp. 1591–1593, 2014.

[91] E. C. Lee and K. R. Park, "Restoration method of skin scattering blurred vein image for finger vein recognition," *Electronics Letters*, vol. 45, no. 21, pp. 1074–1076, 2009.

[92] "The Hong Kong Polytechnic University Finger Image Database version 1.0." [Online]. Available: http://www.comp.polyu.edu.hk/~csajaykr/fvdatabase.htm

[93] Z. Zhang, S. Ma, and X. Han, "Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 4, 2006, pp. 145–148.

[94] D. Wang, J. Li, and G. Memik, "User identification based on finger-vein patterns for consumer electronics devices," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 799–804, 2010.

[95] A. Kumar, "Importance of being unique from finger dorsal patterns: Exploring minor finger knuckle patterns in verifying human identities," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1288–1298, 2014.

[96] S. Aoyama, K. Ito, and T. Aoki, "A multi-finger knuckle recognition system for door handle," in *6th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013, pp. 1–7.

[97] L. Zhang, L. Zhang, D. Zhang, and Z. Guo, "Phase congruency induced local features for finger-knuckle-print recognition," *Pattern Recognition*, vol. 45, no. 7, pp. 2522–2531, 2012.

[98] L. Zhang, L. Zhang, D. Zhang, and H. Zhu, "Ensemble of local and global information for finger–knuckle-print recognition," *Pattern Recognition*, vol. 44, no. 9, pp. 1990–1998, 2011.

[99] A. Kumar and C. Ravikanth, "Personal authentication using finger knuckle surface," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 98–110, 2009.

[100] L. Zhang, L. Zhang, and D. Zhang, "Finger-knuckle-print verification based on band-limited phase-only correlation," in *Computer analysis of images and patterns*. Springer, 2009, pp. 141–148.

[101] L. Zhang, D. Zhang, and D. Zhang, "Finger-knuckle-print: A new biometric identifier," in *16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1981–1984.

[102] L. Zhang, L. Zhang, D. Zhang, and H. Zhu, "Online finger-knuckle-print verification for personal authentication," *Pattern recognition*, vol. 43, no. 7, pp. 2560–2571, 2010.

[103] "The Hong Kong Polytechnic University Finger-Knuckle-Print Database." [Online]. Available: http://www4.comp.polyu.edu.hk/~biometrics/FKP.htm

[104] C. Wang, S. Song, F. Sun, and L. Mei, "Study on finger-articular back texture recognition," in *9th International Conference on Signal Processing*, 2008, pp. 2085–2091.

[105] A. Kumar and Y. Zhou, "Human identification using knucklecodes," in *3rd IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2009, pp. 1–6.

[106] "The Hong Kong Polytechnic University Contactless Finger Knuckle Images Database version 1.0." [Online]. Available: http://www.comp.polyu.edu.hk/~csajaykr/myhome/database_request/fn1/knuckle.htm

[107] R. R. Al-nima, "Design a biometric identification system based on the fusion of hand geometry and backhand patterns," *Iraqi Journal of Statistical Science*, vol. 10, no. 17, pp. 169–180, 2010.

[108] G. Jaswal, A. Kaul, and R. Nath, "Knuckle print biometrics and fusion schemes–overview, challenges, and solutions," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, p. 34, 2016.

[109] L. Zhu and S. Zhang, "Multimodal biometric identification system based on finger geometry, knuckle print and palm print," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1641–1649, 2010.

[110] X. Xu, Q. Jin, L. Zhou, J. Qin, T.-T. Wong, and G. Han, "Illumination-invariant and deformation-tolerant inner knuckle print recognition using portable devices," *Sensors*, vol. 15, no. 2, pp. 4326–4352, 2015.

[111] X.-M. Xu, X.-Z. Lai, Q. Jin, X.-H. Yuan, S.-L. Lai, Y.-W. Lin, and J.-W. Huang, "A novel IKP-based biometric recognition using mobile phone camera," *International Journal of Distributed Sensor Networks*, vol. 2015, p. 2, 2015.

[112] M. Aykut and M. Ekinci, "AAM-based palm segmentation in unrestricted backgrounds and various postures for palmprint recognition," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 955–962, 2013.

[113] Ö. Makul and M. Ekinci, "Biyometric identification based on knuckle prints," in *22nd Signal Processing and Communications Applications Conference (SIU)*, 2014, pp. 1881–1884.

[114] ——, "Palm and palm finger joint surfaces based multibiometric approach," in *39th IEEE International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 665–668.

[115] M. Liu and J. Yan, "Inner-knuckle-print verification based on guided image filtering," in *Proceedings of Chinese Intelligent Automation Conference*. Springer, 2013, pp. 477–484.

[116] M. Liu, Y. Tian, and Y. Ma, "Inner-knuckle-print recognition based on improved LBP," in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*. Springer, 2013, pp. 623–630.

[117] S. Ribaric and I. Fratric, "A biometric identification system based on eigenpalm and eigenfinger features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1698–1709, 2005.

[118] N. Pavesic, S. Ribaric, and B. Grad, "Finger-based personal authentication: a comparison of feature-extraction methods based on principal component analysis, most discriminant features and regularised-direct linear discriminant analysis," *IET Signal Processing*, vol. 3, no. 4, pp. 269–281, 2009.

[119] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[120] H. Ying, T. Tieniu, S. Zhenan, and H. Yufei, "Identity verification by using handprint," in *International Conference on Biometrics*. Springer, 2007, pp. 328–337.

[121] M. K. Goh, C. Tee, and A. B. Teoh, "Bi-modal palm print and knuckle print recognition system," *Journal of IT in Asia*, vol. 3, pp. 53–66, 2010.

[122] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.

[123] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[124] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[125] Q. Tao and R. Veldhuis, "Illumination normalization based on simplified local binary patterns for a face verification system," in *Biometrics Symposium*, 2007, pp. 1–6.

[126] A. Petpon and S. Srisuk, "Face recognition with local line binary pattern," in *5th International Conference on Image and Graphics (ICIG)*, 2009, pp. 533–539.

[127] F. Ahmed, "Gradient directional pattern: A robust feature descriptor for facial expression recognition," *Electronics Letters*, vol. 48, no. 19, pp. 1203–1204, 2012.

[128] L. Nanni and A. Lumini, "A multi-matcher system based on knuckle-based features," *Neural computing and applications*, vol. 18, no. 1, pp. 87–91, 2009.

[129] Y. Zhang, D. Sun, and Z. Qiu, "Hand-based single sample biometrics recognition," *Neural Computing and Applications*, vol. 21, no. 8, pp. 1835–1844, 2012.

[130] K. Woods and K. Bowyer, "Generating ROC curves for artificial neural networks," *IEEE Transactions on Medical Imaging*, vol. 16, no. 3, pp. 329–337, 1997.

[131] R. K. Orr, "Use of a probabilistic neural network to estimate the risk of mortality after cardiac surgery," *Medical Decision Making*, vol. 17, no. 2, pp. 178–185, 1997.

[132] S.-Y. Ooi, A. Teoh, and T.-S. Ong, "Compatibility of biometric strengthening with probabilistic neural network," in *International Symposium on Biometrics and Security Technologies(ISBAST)*, 2008, pp. 1–6.

[133] R. Sharma, A. Singh, A. Joshi, and A. Gangwar, "Robust iris classification through a combination of kernel discriminant analysis and Parzen based probabilistic neural networks," in *4th International Conference of Emerging Applications of Information Technology (EAIT)*, 2014, pp. 268–272.

[134] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[135] A. Joshi, A. Gangwar, R. Sharma, A. Singh, and Z. Saquib, "Periocular recognition based on Gabor and Parzen PNN," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 4977–4981.

[136] N. Almaadeed, A. Aggoun, and A. Amira, "Speaker identification using multimodal neural networks and wavelet analysis," *IET Biometrics*, vol. 4, no. 1, pp. 18–28, 2015.

[137] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, 1997.

[138] L. Huang, H. Zhuang, and W. Zhang, "Person recognition using features of still face images and text-independent speeches," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PacRim*, 2007, pp. 284–287.

[139] S. Meshoul and M. Batouche, "Combining Fisher discriminant analysis and probabilistic neural network for effective on-line signature recognition," in *10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA)*, 2010, pp. 658–661.

[140] J. Hossain and M. Amin, "Leaf shape identification based plant biometrics," in *13th International Conference on Computer and Information Technology (ICCIT)*, 2010, pp. 458–463.

[141] R. Sundaram and B. Dhara, "Neural network based iris recognition system using Haralick features," in *3rd International Conference on Electronics Computer Technology (ICECT)*, vol. 3, 2011, pp. 19–23.

[142] S.-J. Chuang, S.-R. Zeng, and Y.-L. Chou, "Neural networks for the recognition of traditional Chinese handwriting," in *14th IEEE International Conference on Computational Science and Engineering (CSE)*, 2011, pp. 645–648.

[143] L. Sudha and R. Bhavani, "Biometric authorization system by video analysis of human gait in controlled environments," in *International Conference on Recent Trends in Information Technology (ICRTIT)*, 2011, pp. 139–144.

[144] C. Yu and L. Huang, "Biometric recognition by using audio and visual feature fusion," in *International Conference on System Science and Engineering (ICSSE)*, 2012, pp. 173–178.

[145] M. Saini, J. Saini, and S. Sharma, "Moment based wavelet filter design for fingerprint classification," in *International Conference on Signal Processing and Communication (ICSC)*, 2013, pp. 267–270.

[146] P. Wibawa, B. Agung, and F. Sthevanie, "Palm print recognition using competitive hand valley detection, local binary pattern and probabilistic neural network," in *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2014, pp. 105–110.

[147] D. Perwira, B. Agung, and M. Sulistiyo, "Personal palm vein identification using principal component analysis and probabilistic neural network," in *International Conference on Information Technology Systems and Innovation (ICITSI)*, 2014, pp. 99–104.

[148] A. Kumar and S. Shekhar, "Personal identification using multibiometrics rank-level fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 5, pp. 743–752, 2011.

[149] Z. Khan, A. Mian, and Y. Hu, "Contour code: Robust and efficient multispectral palmprint encoding for human recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1935–1942.

[150] Z. Khan, F. Shafait, Y. Hu, and A. Mian, "Multispectral palmprint encoding and recognition," *arXiv preprint arXiv:1402.2941*, 2014.

[151] Y. B. Band, *Light and Matter: Electromagnetism, Optics, Spectroscopy and Lasers*. John Wiley & Sons, 2006, vol. 1.

[152] "The Hong Kong Polytechnic University Low Resolution Fingerprint Database (version 1.0)." [Online]. Available: http://www.comp.polyu.edu.hk/~csajaykr/fplr.htm

[153] A. Kumar and Y. Zhou, "Contactless fingerprint identification using level zero features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011, pp. 114–119.

[154] L. Zhai and Q. Hu, "The research of double-biometric identification technology based on finger geometry & palm print," in *2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2011, pp. 3530–3533.

[155] P. Yu, D. Xu, H. Zhou, and H. Li, "Decision fusion for hand biometric authentication," in *IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, 2009, pp. 486–490.

[156] S. Samavi, F. Kheiri, and N. Karimi, "Binarization and thinning of fingerprint images by pipelining," in *3rd conference on Machine Vision Image Processing and applications-MVIP, University of Tehran, Iran*, vol. 2, 2005.

[157] A. Mukherjee and S. Kanrar, "Enhancement of image resolution by binarization," *arXiv preprint arXiv:1111.4800*, 2011.

[158] MATLAB, *Image Processing Toolbox, For Use with MATLAB, Computation, Visualization, Programming. version 3*. Natick, MA: The MathWorks Inc., 2001.

[159] D. P. Hader, Ed., *Image analysis: methods and applications*, second edition ed. CRC PRESS, 1992.

[160] N. Jamil, T. M. T. Sembok, and Z. Bakar, "Noise removal and enhancement of binary images using morphological operations," in *International Symposium on Information Technology, ITSim*, vol. 4, 2008, pp. 1–6.

[161] S. M. Hsu, "Extracting target features from angle-angle and range-Doppler images," DTIC Document, Tech. Rep., 1993.

[162] MATLAB, *Image Processing Toolbox*. The MathWorks Inc., 1994-2012.

[163] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[164] D.-J. Kim, S.-H. Lee, and M.-K. Sohn, "Face recognition via local directional pattern," *International Journal of Security and Its Applications*, vol. 7, no. 2, pp. 191–200, 2013.

[165] O. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," in *Proceedings of Informing Science & IT Education Conference (InSITE)*, vol. 40, 2009, pp. 97–107.

[166] A. Ukil, V. H. Shah, and B. Deck, "Fast computation of arctangent functions for embedded applications: A comparative analysis," in *IEEE International Symposium on Industrial Electronics*, 2011, pp. 1206–1211.

[167] T. Huang, J. Burnett, and A. Deczky, "The importance of phase in image processing filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 6, pp. 529–542, 1975.

[168] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.

[169] L. Junli, Y. Gengyun, and Z. Guanghui, "Evaluation of tobacco mixing uniformity based on chemical composition," in *31st Chinese Control Conference (CCC)*, 2012, pp. 7552–7555.

[170] M. G. K. Ong, C. Tee, and A. T. B. Jin, "Touch-less palm print biometric system," in *International Conference on Computer Vision Theory and Applications*, 2008.

[171] M. Hanmandlu, A. Gureja, and A. Jain, "Palm print recognition using local binary pattern operator and support vector machines," in *International Conference on Signal and Image Processing (ICSIP)*, 2010, pp. 158–162.

[172] I. T. Young, J. J. Gerbrands, and L. J. Van Vliet, *Fundamentals of Image Processing*. Delft University of Technology Delft, The Netherlands, 1998.

[173] B. Topçu and H. Erdogan, "Decision fusion for patch-based face recognition," in *20th IEEE International Conference on Pattern Recognition (ICPR)*, 2010, pp. 1348–1351.

[174] O. Vatamanu and M. Jivulescu, "Image classification using local binary pattern operators for static images," in *8th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2013, pp. 173–178.

[175] L. V. Fausett and P. Hall, *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall Englewood Cliffs, 1994.

[176] A. Ross and R. Govindarajan, "Feature level fusion in biometric systems," in *proceedings of Biometric Consortium Conference (BCC)*, 2004.

[177] A. A. Ross and R. Govindarajan, "Feature level fusion of hand and face biometrics," in *Defense and Security*. International Society for Optics and Photonics, 2005, pp. 196–204.

[178] A. Gupta, E. Walia, and M. Vaidya, "Feature level fusion of palm print and fingerprint modalities using discrete cosine transform," in *International Conference on Advances in Engineering and Technology Research (ICAETR)*, 2014, pp. 1–5.

[179] W. Woo and S. Dlay, "Regularised nonlinear blind signal separation using sparsely connected network," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 152, no. 1, pp. 61–73, 2005.

[180] J. Kou, S. Xiong, S. Wan, and H. Liu, "The incremental probabilistic neural network," in *6th International Conference on Natural Computation (ICNC)*, vol. 3, 2010, pp. 1330–1333.

[181] M. Hajmeer and I. Basheer, "A probabilistic neural network approach for modeling and classification of bacterial growth/no-growth data," *Journal of Microbiological Methods*, vol. 51, no. 2, pp. 217–226, 2002.

[182] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[183] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.